



Trajectory generation algorithm for safe human-robot collaboration based on multiple depth sensor measurements

Matteo Ragaglia^{*,a,b}, Andrea Maria Zanchettin^a, Paolo Rocco^a

^a Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria, Piazza Leonardo Da Vinci 32, Milano, Italy

^b Yanmar R&D Europe, Viale Galileo 3/A, Firenze, Italy



ARTICLE INFO

Keywords:

Human-robot collaboration
Safety
Trajectory generation
Motion tracking
Depth sensors

ABSTRACT

Computing and modifying in real-time the trajectory of an industrial robot involved in a Human-Robot Collaboration (HRC) scenario is a challenging problem, mainly because of two conflicting requirements: ensuring the human worker's safety and completing the task assigned to the robot. This paper proposes a novel trajectory generation algorithm conceived to maximize productivity while taking into account safety requirements as actual constraints. At first, safety constraints are formulated by taking into account a manipulator and a set of arbitrarily-shaped convex obstacles. Then, a sensor fusion algorithm merges together the measurements acquired from different depth sensors and outputs a noise-free estimation of the kinematic configuration of a human worker moving inside the robotic cell. This estimation is then used to predict the space that the human worker can occupy within the robot stopping time in terms of a set of convex swept volumes. By considering these swept volumes as obstacles, the robot controller can modify the pre-programmed trajectory in order to enforce the safety constraints (thus avoiding collision with the human worker), while preventing task interruption. The proposed trajectory generation algorithm is validated through several experiments performed on an ABB IRB140 industrial robot.

1. Introduction

Industrial robots are able to offer fast and accurate task execution in various industrial applications like for instance: welding, painting, packaging, palletizing, etc. Nevertheless they are still rarely used within Small and Medium-sized Enterprises (SMEs), mainly because installation, setup and programming of a robotized workstation are time-consuming activities that require a lot of skilled engineering effort. Moreover, industrial robots usually need to be separated from the human workspace by physical barriers [1] for safety reasons. Given these limitations and considering also limited budget and constraints on space consumption, it is clear why the large majority of SMEs cannot afford to extensively use industrial robots. On the other hand the elimination of physical barriers to allow direct Human-Robot Collaboration (HRC) surely represents the key factor that will facilitate industrial robots to be massively used also in SMEs.

However, HRC requires a balance between safety and productivity: the human worker should not be harmed by the robot (see [2,3]), but also production constraints (i.e. programmed task requirements) should not be violated, in order to preserve the manipulator's productivity. Recently several contributions to the problem of planning and control

of industrial manipulators for safe HRC have been proposed. For instance [4] describes a passivity based controller that produces slower motion profiles when the human/robot distance decreases. Different approaches relying on evasive motion in order to avoid human-robot collisions have also been proposed, ranging from potential fields [5] to velocity obstacles [6], Danger Field [7–9], depth spaces [10] and Safety Field [11].

On the other hand, several collision detection (see [12]) and reaction strategies have also been proposed, like for instance [13,14]. In particular [15] discusses a comparison between different collision reaction strategies aiming at ensuring human safety in physical-HRC scenarios, while [16] and [17] introduce real-time self collision detection algorithms for both humanoid and industrial robots, based on the concept of “swept volumes”. The most relevant safety standards published in the last few years [18,19] introduced “speed and separation monitoring” criteria, according to which a minimum separation distance (depending on Tool Center Point velocity and/or payload) must be kept between an industrial manipulator and a human worker. Even though the most recent safety standard [20] allows physical contact between the robot and the human worker, there are several scenarios in which human-robot collisions should be avoided. For

* Corresponding author.

E-mail addresses: matteo_ragaglia@yanmar.com (M. Ragaglia), andreamaria.zanchettin@polimi.it (A.M. Zanchettin), paolo.rocco@polimi.it (P. Rocco).

instance, the robot should never hit the human worker's head and collisions involving heavy-payload manipulators should be always avoided.

Several control policies that are compliant to these criteria have been proposed recently, like for instance in [21–23]. In particular, [24–26] proposed various control algorithms that rely on kinematic scaling in order to satisfy the “minimum separation distance” criterion. Clearly, to ensure and improve the effectiveness of these control strategies it is necessary to predict “human occupancy”, i.e. the space occupied by the human during a certain amount of time. In the last few years several contributions have focused on predicting human motion, mainly in terms of forecasting the final destination of a human walking trajectory [27,28], but also in terms of estimating human occupancy [29,30]. Finally, the application of these prediction algorithms for control purposes has been investigated in [31–33], mainly in terms of pre-collision strategies. For instance, in [34] the authors exploit a prediction of human occupancy (obtained by merging multiple depth sensor measurements) together with a forward physical simulation of the robot in order to stop its motion whenever an imminent collision is detected.

The main contributions of this work can be listed as follows:

- a **sensor fusion algorithm** that merges together the measurements acquired from different depth sensors to estimate human joint positions and velocities;
- a novel strategy to **predict human space occupancy** that results in a much more accurate and significantly less conservative prediction with respect to state of the art alternatives;
- a **reactive trajectory generation algorithm** that allows the manipulator to deviate from a pre-programmed path in order to satisfy minimum separation distance constraints.

To sum up, the whole system works in the following way. Considering the aforementioned human kinematic model, a set of joint positions can be extracted from 3-dimensional skeletal points acquired from several depth sensors. These measurements are merged together by the sensor fusion algorithm that outputs the estimation of the human joint positions and velocities. Differently from the approaches introduced in [24] and [26], not only the proposed sensor fusion algorithm relies on more than one sensor, but it also allow to robustly estimate the kinematic configuration of the human worker in presence of measurement noise, sensor failures, outliers and occlusions.

The estimated kinematic configuration allows to predict the space occupancy of the human worker within the robot worst-case braking time (see [35]). This prediction consists in a series of swept volumes that, differently from what was originally proposed in [26], are computed on the basis of both human joint positions and velocities. This solution allows to obtain a much more accurate and significantly less conservative prediction of human occupancy, particularly for slow motions. Since this predicted occupancy consists in a series of convex swept volumes, the minimum distance criterion can be formulated for each volume in terms of “safety constraints”, as they were introduced in [24] and further refined in [25] and [26].

Finally, constrained optimization is employed in order to ensure that the robot motion satisfies these safety constraints, while minimizing deviations from the pre-programmed trajectory. With respect to the kinematic scaling algorithms proposed in [24–26], this solution allows to significantly reduce the amount of time during which the manipulator is still, thus improving its efficiency without violating the safety constraints.

The paper is organized as follows. Section 2 summarizes our previous work regarding safety constraints, while the adopted human kinematic model and the occupancy prediction strategy are described in Section 3. Section 4 focuses on the sensor fusion algorithm that estimates the kinematic state of the human worker by merging together multiple depth sensor measurements. This estimate acts as the main

input of the trajectory generation algorithm introduced in Section 5. Section 6 presents the case study against which our proposed method has been tested and analyzes the results of several validation experiments. Finally, Section 7 compares the proposed approach with state-of-the-art techniques, while concluding remarks and future work directions can be found in Section 8.

2. Background on safety constraints

One of the fundamental elements of the proposed trajectory generation algorithm is the concept of “safety constraints”, i.e. a set of mathematical constraints guaranteeing that, in presence of one or more detected obstacles, the robot is always able to stop before colliding with the environment. In other words, the robot trajectory must obey, at all time, the minimum separation distance criterion, defined as:

$$\text{velocity} \cdot T_s \leq \max(0, \text{distance} - \text{clearance}) \quad (1)$$

where “velocity” represents the robot speed in the direction of the obstacle, the worst-case braking time T_s possibly depends on the robot payload [1], “distance” is the distance between the robot and a generic obstacle (possibly a human worker) and the “clearance” parameter allows to take into account both robot and obstacle dimensions, sensor uncertainties and ultimately an actual clearance. For the sake of clearness, a visual representation of the minimum separation distance criterion is shown in Fig. 1, where an example of the here proposed prediction of human occupancy is shown. It is worth mentioning that an alternative, yet equivalent, interpretation of inequality (1) consists in stating that the robot speed must be adapted on the basis of the distance with respect to the obstacles, hence leading to the speed and separation monitoring approach. In the following we report some background material on the derivation of the safety constraints. Starting from inequality (1), we first formulate the safety constraints for the case of

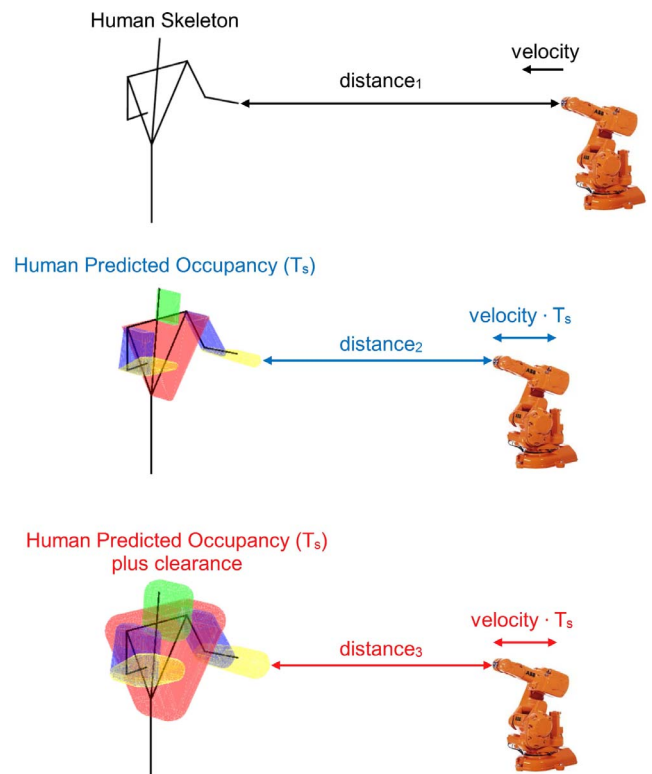


Fig. 1. Image representation of the minimum separation distance criterion. Top - distance_1 : separation distance between the human skeleton and the robot. Mid - distance_2 : separation distance between the human occupancy (predicted within T_s) and the robot moving at velocity for T_s s. Bottom - distance_3 : separation distance between the human occupancy, plus the *clearance* parameter, and the moving robot.

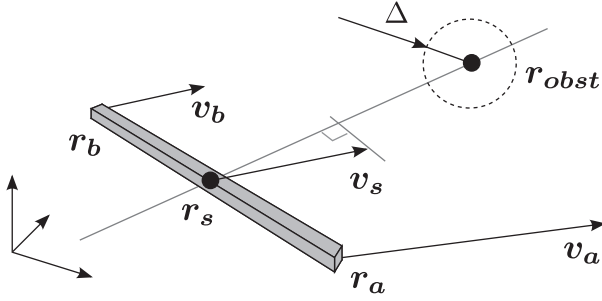


Fig. 2. A rigid beam representing one link.

point-shaped obstacles (as introduced in [24]) and then we generalize them for arbitrarily-shaped convex obstacles (as proposed in [26]).

2.1. Safety constraints for point-shaped obstacles

Consider Fig. 2 which shows a point obstacle r_{obst} as well as a generic robotic link, represented as a rigid beam whose endpoints are at positions r_a and r_b .

Defining r_s a generic point on the robot link (with velocity v_s), the safety requirement (1) can be rewritten in the form of the following inequality:

$$v_s^T \frac{r_{obst} - r_s}{\|r_{obst} - r_s\|} T_s \leq \max(0, \|r_{obst} - r_s\| - \Delta) \quad (2)$$

where Δ is a clearance parameter. This constraint can be further arranged as

$$v_s^T (r_{obst} - r_s) T_s \leq \max(0, \|r_{obst} - r_s\|^2 - \Delta \|r_{obst} - r_s\|) \quad (3)$$

Assume now the following parametrization of the link in terms of positions and velocities of its end points

$$r_s = r_a + s(r_b - r_a) \quad v_s = v_a + s(v_b - v_a) \quad (4)$$

where $s \in [0, 1]$ is a natural coordinate. In order to enforce the safety constraints, we require the inequality in (3) to be satisfied for all $s \in [0, 1]$. The left hand side becomes

$$v_s^T (r_{obst} - r_s) = v_a^T (r_{obst} - r_a) + s(v_b - v_a)^T (r_{obst} - r_a) - s v_a^T (r_b - r_a) - s^2 \underbrace{(v_b - v_a)^T (r_b - r_a)}_{=0} \quad (5)$$

Notice that vector $(v_b - v_a)$ represents the time derivative of vector $(r_b - r_a)$, whose norm is invariant with respect to time. Consequently, the two vectors are perpendicular and the coefficient of s^2 is null.

As for the right hand side, notice that

$$[\max(0, \|r_{obst} - r_s\| - \Delta)]^2 \leq \max(0, \|r_{obst} - r_s\|^2 - \Delta \|r_{obst} - r_s\|) \quad (6)$$

As a result, the set of inequalities describing the safety constraints can be written as follows

$$\alpha + \beta s \leq g(s), \quad \forall s \in [0, 1] \quad (7)$$

where

$$\begin{aligned} \alpha &= e T_s v_a^T (r_{obst} - r_a) \\ \beta &= T_s (v_b - v_a)^T (r_{obst} - r_a) - T_s v_a^T (r_b - r_a) \\ g(s) &= [\max(0, \|r_{obst} - r_s\| - \Delta)]^2 \end{aligned} \quad (8)$$

By noticing that the left hand side is a linear function in s , it is possible to write the following sufficient condition for the safety constraint (7) to be satisfied

$$\max\{\alpha, \alpha + \beta\} \leq \min_s g(s) \quad (9)$$

In turn, within the right hand side term, it is possible to exchange the min and max operators, obtaining

$$\min_s g(s) = \left[\max\left(0, \min_s \|r_{obst} - r_s\| - \Delta\right) \right]^2 \quad (10)$$

where the term $\min_s \|r_{obst} - r_s\| - \Delta$ represents, when it is positive, the distance between a sphere of radius Δ centered in r_{obst} and the segment from r_a to r_b . Finally, we obtain the following pair of inequalities

$$\begin{aligned} \alpha &= T_s (r_{obst} - r_a)^T v_a \leq \min_s g(s) \\ \alpha + \beta &= T_s (r_{obst} - r_a)^T v_b - T_s (r_b - r_a)^T v_a \leq \min_s g(s) \end{aligned} \quad (11)$$

Summarizing, in case of point-shaped obstacles, the minimum separation distance criterion can be written in matrix form as $T_s E \dot{q} \leq f$ where

$$\begin{aligned} E &= \begin{bmatrix} (r_{obst} - r_a)^T J_a \\ (r_{obst} - r_a)^T J_b - (r_b - r_a)^T J_a \end{bmatrix} \\ f &= \min_s g(s) \begin{bmatrix} 1 \\ 1 \end{bmatrix} \end{aligned} \quad (12)$$

J_a and J_b are position Jacobians of the two link end points. It is worth mentioning that possible rotational velocities of the link are ignored in the formalization of the safety constraints, since they do not impact on the distance between the beam and the obstacles.

2.2. Safety constraints for arbitrarily-shaped convex obstacles

In order to extend the formulation of the safety constraints to obstacles having more complex geometry, let us consider a generic polytopic obstacle \mathcal{O} , as shown in Fig. 3. The constraints to be enforced for such an obstacle can be written as follows

$$T_s E(r_{obst}) \dot{q} \leq f(r_{obst}), \quad \forall r_{obst} \in \mathcal{O} \quad (13)$$

The number of constraints to be enforced at run time is conceptually infinite, i.e. one per each point belonging to \mathcal{O} . However, some geometrical properties of the obstacle can be exploited in order to make the problem tractable. A sufficient condition for (13) to be satisfied for all points $r_{obst} \in \mathcal{O}$ is

$$T_s E(r_{obst}) \dot{q} \leq d \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \forall r_{obst} \in \mathcal{O} \quad (14)$$

where the right hand side term $d = \min_{r_{obst} \in \mathcal{O}} \|f(r_{obst})\|_\infty$ represents the minimum distance between the link of the robot and the polytopic obstacle \mathcal{O} . It can be easily computed using the GJK algorithm, [36]. Moreover, notice from (12) that the left hand side term is linear with respect to the parameter $r_{obst} \in \mathcal{O}$. Therefore the safety constraints regarding the pair link-obstacle can be written as follows

$$E_0 = \begin{bmatrix} J_a \\ J_b \end{bmatrix} \quad E_1 = \begin{bmatrix} -r_a^T J_a \\ -r_a^T J_b - (r_b - r_a)^T J_a \end{bmatrix} \quad (15)$$

$$T_s (r_{obst}^T E_0 + E_1) \dot{q} \leq d, \quad \forall r_{obst} \in \mathcal{O} \quad (16)$$

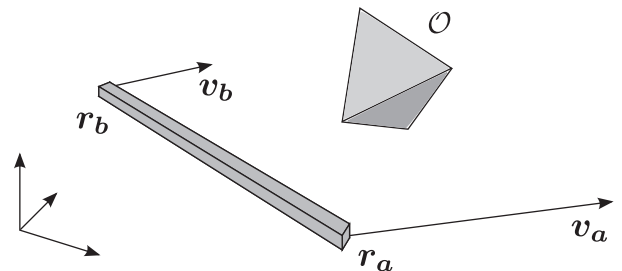


Fig. 3. A generic polytopic (convex) obstacle.

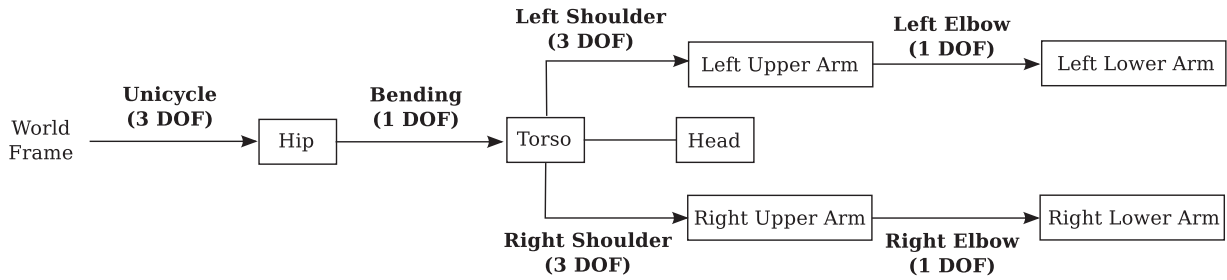


Fig. 4. Kinematic model of the human: DOFs, frames and bodies.

For linearity (and thus convexity) the aforementioned constraint (which actually still consists of an infinite number of scalar inequalities) can be equivalently written in terms of the vertices (thus a limited number) of the polytope representing the obstacle \mathcal{O} , hence $\forall \mathbf{r}_{\text{obst}} \in \text{vert}(\mathcal{O})$.

3. Human kinematics and prediction of occupancy

In the previous Section, we presented a mathematical formulation to represent the safety constraint arising in a typical human-robot collaboration scenario. As already discussed, even if it is possible to consider arbitrarily geometrically shaped obstacles, their motion is not directly accounted for in the expression of the safety requirement. However, in a HRC setup, the motion of the human, and particularly the prediction of his/her occupancy, has to be clearly taken into account to safely adjust the trajectory of the robot.

In order to compute this prediction, we make use of a relatively simple kinematic model suitable for real-time calculations. The model of the human, see Fig. 4, consists of a mobile base that represents the walking kinematics, one lumped one-DOF (flexion/extension) torso, a fixed head and two four-DOF arms. The main objective of this Section is to introduce a simple algorithm that predicts a volume around the human silhouette, which can be possibly reached in T_s seconds, i.e. within the time needed by the robot to stop.

In the following, further details regarding the kinematics of walking, as well as the kinematic model of the human arm, are given. The predicted reachable area will be represented in terms of superposition of each possible motion (swept motion) in the corresponding prediction horizon.

3.1. Modelling the kinematics of the human walk

For the description of the kinematics of walking, as suggested in e.g. [27,37], we make use of the unicycle model

$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{\theta} = \omega \end{cases} \quad (17)$$

together with the following constraints on input velocities:

$$\begin{aligned} 0 &\leq v \leq v^{\text{sup}} \\ \omega^{\text{inf}} &\leq \omega \leq \omega^{\text{sup}} \end{aligned} \quad (18)$$

where x , y and θ describe the walking human pose with respect to the world base frame, v is the linear velocity and ω is the angular velocity. We explicitly exclude the possibility of walking backward, since it is a quite unsafe and uncommon behaviour in an industrial environment.

In order to compute the human walking reachable set in T_s seconds, at first we consider a generic starting configuration:

$$\begin{aligned} x(0) &= x_0, & y(0) &= y_0, & \theta(0) &= \theta_0 \\ v(0) &= v_0, & \omega(0) &= \omega_0 \end{aligned} \quad (19)$$

Moreover, in addition to constraints (18), we impose fixed lower and upper bounds on linear and angular accelerations: \dot{v}^{inf} , \dot{v}^{sup} , $\dot{\omega}^{\text{inf}}$ and

$\dot{\omega}^{\text{sup}}$, respectively. Consequently, it is possible to compute the minimum and maximum velocities that a walking human can reach within T_s seconds:

$$\begin{aligned} v^{\text{min}} &= 0 \\ v^{\text{max}} &= \min\{v_0 + \dot{v}^{\text{sup}} \cdot T_s, v^{\text{sup}}\} \\ \omega^{\text{min}} &= \max\{\omega_0 + \dot{\omega}^{\text{inf}} \cdot T_s, \omega^{\text{inf}}\} \\ \omega^{\text{max}} &= \min\{\omega_0 + \dot{\omega}^{\text{sup}} \cdot T_s, \omega^{\text{sup}}\} \end{aligned}$$

Similarly to the calculations presented in [26], these minimum and maximum velocities can be used to determine a conservative approximation of the human walking T_s -reachable set in terms of two distinct circular sectors of radius $R = v^{\text{max}} T_s$, spanning two different angles β^{min} and β^{max} with respect to the original direction of motion θ_0 , as shown in Fig. 5. In particular:

$$\beta^{\text{min}} = 2 \cos^{-1} \left(\frac{\sin(\omega^{\text{min}} T_s)}{\sqrt{2 - 2 \cos(\omega^{\text{min}} T_s)}} \right) \quad (20)$$

$$\beta^{\text{max}} = 2 \cos^{-1} \left(\frac{\sin(\omega^{\text{max}} T_s)}{\sqrt{2 - 2 \cos(\omega^{\text{max}} T_s)}} \right) \quad (21)$$

In this way, the human walking model can be described as a composition of two independent degrees of freedom (DOFs): a rotational one followed by a translational one. The corresponding reachable set follows immediately from this approximation:

$$\mathbf{x}_{T_s} = \begin{bmatrix} x_{T_s}^- \\ x_{T_s}^+ \end{bmatrix} = \begin{bmatrix} x_0 \\ x_0 + v^{\text{max}} T_s \cos \theta_0 \end{bmatrix} \quad (22)$$

$$\mathbf{y}_{T_s} = \begin{bmatrix} y_{T_s}^- \\ y_{T_s}^+ \end{bmatrix} = \begin{bmatrix} y_0 \\ y_0 + v^{\text{max}} T_s \sin \theta_0 \end{bmatrix} \quad (23)$$

$$\theta_{T_s} = \begin{bmatrix} \theta_{T_s}^- \\ \theta_{T_s}^+ \end{bmatrix} = \begin{bmatrix} \theta_0 + \beta^{\text{min}} \\ \theta_0 + \beta^{\text{max}} \end{bmatrix} \quad (24)$$

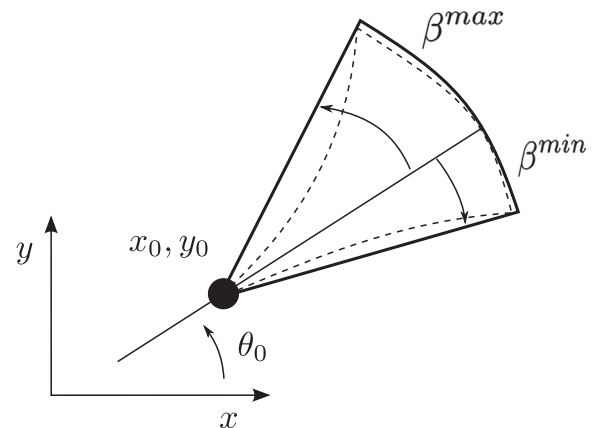


Fig. 5. Unicycle reachable set (dashed) and its convex approximation (solid bold).

Finally, in order to account for lateral walking, it is possible to extend model (17) according to [38]:

$$\begin{cases} \dot{x} = v \cos \theta - v_{\perp} \sin \theta \\ \dot{y} = v \sin \theta + v_{\perp} \cos \theta \\ \dot{\theta} = \omega \end{cases} \quad (25)$$

$$v_{\perp}^{inf} \leq v_{\perp} \leq v_{\perp}^{sup} \quad (26)$$

where v_{\perp} is a velocity component that is orthogonal with respect to the forward direction of the human defined by θ .

Given an initial orthogonal velocity $v_{\perp,0}$ and introducing v_{\perp}^{inf} and v_{\perp}^{sup} as lower and upper bounds on orthogonal acceleration, the minimum and maximum orthogonal velocities that a walking human can reach within T_s seconds are:

$$v_{\perp}^{min} = \max\{v_{\perp,0} + \dot{v}_{\perp}^{inf} T_s, v_{\perp}^{inf}\} \quad (27)$$

$$v_{\perp}^{max} = \min\{v_{\perp,0} + \dot{v}_{\perp}^{sup} T_s, v_{\perp}^{sup}\} \quad (28)$$

Consequently, the x_{T_s} and y_{T_s} components of the reachable set determined in (22) can be modified as follows:

$$x_{T_s} \leftarrow x_{T_s} + \begin{bmatrix} -v_{\perp}^{min} T_s \sin \theta_0 \\ -v_{\perp}^{max} T_s \sin \theta_0 \end{bmatrix} \quad (29)$$

$$y_{T_s} \leftarrow y_{T_s} + \begin{bmatrix} v_{\perp}^{min} T_s \cos \theta_0 \\ v_{\perp}^{max} T_s \cos \theta_0 \end{bmatrix} \quad (30)$$

Finally, it is worth noticing that model (25) can be expressed in terms of a linear formulation by simply considering fully de-coupled linear velocities v_x and v_y :

$$\begin{cases} \dot{x} = v_x \\ \dot{y} = v_y \\ \dot{\theta} = \omega \\ \dot{v} = a_l \\ \dot{\omega} = a_{\omega} \end{cases} \quad (31)$$

As a matter of fact, the new formulation is completely equivalent to the previous one, since a unique correspondence exists between the two different sets of velocities:

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} v \\ v_{\perp} \end{bmatrix} \quad (32)$$

$$\begin{bmatrix} v \\ v_{\perp} \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} \quad (33)$$

3.2. Modelling the human arm kinematics

As for the human arm, we make use of a well-known model, sketched in Fig. 6, comprising one spherical joint representing the shoulder and one additional DOF for the elbow, see e.g. [39,40]. Moreover, we add a rotational joint modelling the flexion/extension of the torso before the arm kinematic chain, i.e. the ρ joint angle in Fig. 6.

The workspace of the human arm is clearly limited due to some intrinsic limitations in the gleno-humeral joint (shoulder) as well as in the elbow. Differently from robots, however, these limits are coupled, as described e.g. in [41]. In this work, the authors identify the admissible range of motion of a healthy human arm as the region identified by the following constraints:¹

$$-9 \leq \alpha_1 \leq 160 \quad (34a)$$

$$-43 + \frac{\alpha_1}{3} \leq \alpha_2 \leq 153 - \frac{\alpha_1}{6} \quad (34b)$$

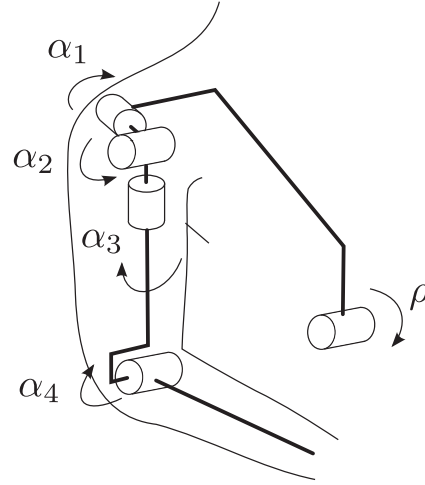


Fig. 6. Kinematic model of the human (right) arm and torso flexion/extension angle ρ .

$$-90 + \frac{7\alpha_1}{9} - \frac{\alpha_2}{9} + \frac{2\alpha_1\alpha_2}{810} \leq \alpha_3 \leq 60 + \frac{4\alpha_1}{9} - \frac{5\alpha_2}{9} + \frac{5\alpha_1\alpha_2}{810} \quad (34c)$$

$$20 \leq \alpha_4 \leq 180 \quad (34d)$$

where $\alpha_1, \dots, \alpha_4$ are the arm joint angles shown in Fig. 6. As for the flexion/extension of the torso we consider the following bound

$$-30 \leq \rho \leq 90 \quad (35)$$

Given a kinematic configuration (i.e. α and $\dot{\alpha}$) and worst-case, thus conservative, acceleration bounds $\ddot{\alpha}^{sup}$ and $\ddot{\alpha}^{inf}$, the reachable set in T_s seconds

$$\alpha_{T_s} = [\alpha_{T_s}^{-T}, \alpha_{T_s}^{+T}]^T \quad (36)$$

can be computed according to Algorithm 1. Given an arm joint, to compute its maximum reachable value (lines 2–7), the maximum acceleration is uniformly applied for T_s s (line 3). In case the maximum joint velocity is reached, the corresponding instant T^* is identified (line 5) and a null acceleration is applied from T^* to T_s (line 6). The same procedure is used to determine the minimum reachable value of the arm joint (lines 8–13).

Identical considerations lead to the calculation of $\rho_{T_s} = [\rho_{T_s}^-, \rho_{T_s}^+]^T$. Finally, we limit the obtained result to the region inside the joint angle limits, previously introduced in (34) and (35).

3.3. Prediction and representation of the human occupancy

In the following we present a method to predict the occupancy of the human silhouette in terms of a series of convex polytopes, using the computation of the reachable sets for each DOF. Assume a given configuration of the human upper body:

$$\begin{aligned} p &= [x \ y \ \theta \ \rho \ \alpha^{right} \ \alpha^{left}]^T \\ \dot{p} &= [v_x \ v_y \ \omega \ \dot{\rho} \ \dot{\alpha}^{right} \ \dot{\alpha}^{left}]^T \end{aligned} \quad (37)$$

where \dot{x} and \dot{y} are defined according to (17), while α^{left} and α^{right} are the joint angles vectors for the left and right arm, respectively. Once we determine, for each DOF, the T_s -reachable set lower and upper bounds

$$p_{T_s}^- = [x_{T_s}^-, y_{T_s}^-, \theta_{T_s}^-, \rho_{T_s}^-, \alpha_{T_s}^{left-}, \alpha_{T_s}^{right-}]^T \quad (38)$$

$$p_{T_s}^+ = [x_{T_s}^+, y_{T_s}^+, \theta_{T_s}^+, \rho_{T_s}^+, \alpha_{T_s}^{left+}, \alpha_{T_s}^{right+}]^T \quad (39)$$

the problem of computing and representing the human occupancy can be solved by mapping the following inequalities into a corresponding region inside the 3D Cartesian space:

¹ All listed parameters are referred to angles in degrees.

```

1: for all  $\alpha_i \in \alpha$  do
2:   if  $\dot{\alpha}_i + \ddot{\alpha}_i^{sup} T_s \leq \dot{\alpha}_i^{sup}$  then
3:      $\alpha_{T_s,i}^+ = \alpha_i + \dot{\alpha}_i T_s + \frac{1}{2} \ddot{\alpha}_i^{sup} T_s^2$ 
4:   else
5:      $T^* = (\dot{\alpha}_i^{sup} - \dot{\alpha}_i) / \ddot{\alpha}_i^{sup}$ 
6:      $\alpha_{T_s,i}^+ = \alpha_i + \dot{\alpha}_i T^* + \frac{1}{2} \ddot{\alpha}_i^{sup} (T^*)^2 + \dot{\alpha}_i^{sup} (T_s - T^*)$ 
7:   end if
8:   if  $\dot{\alpha}_i + \ddot{\alpha}_i^{inf} T_s \geq \dot{\alpha}_i^{inf}$  then
9:      $\alpha_{T_s,i}^- = \alpha_i + \dot{\alpha}_i T_s + \frac{1}{2} \ddot{\alpha}_i^{inf} T_s^2$ 
10:  else
11:     $T^* = (\dot{\alpha}_i^{inf} - \dot{\alpha}_i) / \ddot{\alpha}_i^{inf}$ 
12:     $\alpha_{T_s,i}^- = \alpha_i + \dot{\alpha}_i T^* + \frac{1}{2} \ddot{\alpha}_i^{inf} (T^*)^2 + \dot{\alpha}_i^{inf} (T_s - T^*)$ 
13:  end if
14: end for

```

Algorithm 1. Human arm T_s -reachable set α_{T_s} .

$$\min(\mathbf{p}_{T_s}^-, \mathbf{p}) \leq \mathbf{p} \leq \max(\mathbf{p}_{T_s}^+, \mathbf{p}) \quad (40)$$

where the $\min(\cdot)$ and the $\max(\cdot)$ functions are applied element-wise in order to guarantee that, for each DOF, all the possible values ranging from the start configuration to the T_s -reachable set are actually considered.

Given a generic 3-dimensional convex object \mathcal{O} , there are two different swept volumes that can be computed: a translational swept volume and a rotational one, depending on the kind of motion \mathcal{O} is subjected to. If we consider a prismatic joint, the translational swept volume of \mathcal{O} can be computed by applying the corresponding translation to each point belonging to \mathcal{O} and by determining the convex hull of the resulting points, as shown in Fig. 7.

In turn, for a rotational DOF, we exploit one of the methods described in [16]. In particular, when applying a rotation to each point belonging to \mathcal{O} , we obtain a circular arc. A possible way to approximate this arc with a finite number of points is to construct a triangle as it is shown in [16, Fig. 3(c)]. Therefore, the rotational swept volume of \mathcal{O} can be easily obtained as the convex hull of the vertices of all the triangles, see Fig. 8.

Knowing how to determine the translational and rotational swept volumes for a generic set of points, the prediction of human occupancy can be determined by computing a specific swept volume for each limb: head (HD), thorax (THX), upper left arm (from shoulder to elbow - ULA), lower left arm (from elbow to wrist - LLA), upper right arm (URA) and lower right arm (LRA). Assume that a convex object \mathcal{V}_l representing the l th limb is given by means of its vertices set. Then, referring to the tree describing the human kinematic model in Fig. 4, a list of DOFs from the current limb to the world frame can be arranged.

Consequently, the swept volume of the l th limb can be determined by iteratively applying the proper sweeping strategy to \mathcal{V}_l for each DOF connecting the l th limb to the world-base frame. Each sweeping operation is calculated on the basis of the upper and lower bounds $\mathbf{p}_{T_s}^-$ and $\mathbf{p}_{T_s}^+$ previously computed (see Sections 3.1 and 3.2). A pseudo-code version of this procedure is sketched in Algorithm 2, where:

- $\text{kinematicChain}(l, \mathbf{p})$ is a function that computes \mathcal{P} , i.e. the set of DOFs connecting the limb to the world-base frame. The set is ordered starting from the limb and going backwards along the kinematic chain until the world-base frame is reached;
- $\text{InitSweptVolume}()$ is a function that initializes a swept volume by including into the set the endpoints of the link corresponding to the first DOF;
- $\text{isPrismatic}(p_i)$ is a logical predicate that is true if its argument is a

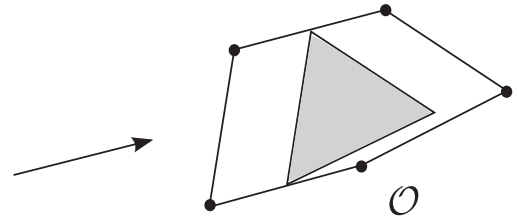


Fig. 7. Translational swept volume of convex objects and its convex hull.

prismatic DOF and false otherwise;

- $\mathbf{p}_{T_s,i}^-$ and $\mathbf{p}_{T_s,i}^+$ are, respectively, the lower and upper bound of the i -th DOF selected by kinematicChain ;
- $\text{sweepLin}(\mathcal{V}, \mathbf{p}_{T_s,i}^-, \mathbf{p}_{T_s,i}^+)$ computes the translational swept volume of the set of points \mathcal{V} , given the DOF bounds;
- $\text{sweepRot}(\mathcal{V}, \mathbf{p}_{T_s,i}^-, \mathbf{p}_{T_s,i}^+)$ computes the rotational swept volume of the set of points \mathcal{V} , given the DOF bounds;
- A_{i-1}^i is the linear transformation matrix from frame i to frame $i - 1$, with respect to the output of kinematicChain ;
- \mathbf{r}_l^j is the j th vertex of the l th swept volume \mathcal{V}_l .

The resulting representation of human occupancy is shown in Fig. 9. Finally, in order to account for the dimensions of the different human

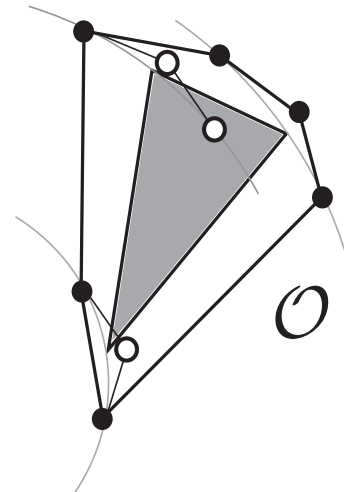


Fig. 8. Rotational swept volume of convex objects and its convex hull.

```

1:  $\mathcal{SV} \leftarrow \emptyset$ ;
2:  $L = \{HD, THX, ULA, LLA, URA, LRA\}$ ;
3: for all  $l \in L$  do
4:    $\mathcal{P} = \text{kinematicChain}(l, \mathbf{p})$ ;
5:    $\mathcal{V}_l \leftarrow \text{InitSweptVolume}()$ ;
6:   if  $\text{isPrismatic}(p_1)$  then
7:      $\mathcal{V}_l \leftarrow \text{sweepLin}(\mathcal{V}_l, p_{T_{s,i}}^-, p_{T_{s,i}}^+)$ ;
8:   else
9:      $\mathcal{V}_l \leftarrow \text{sweepRot}(\mathcal{V}_l, p_{T_{s,i}}^-, p_{T_{s,i}}^+)$ ;
10:  end if
11:  for all  $p_i \in \mathcal{P}$  do
12:    if  $p_i \neq p_1$  then
13:       $\mathcal{V}_l \leftarrow A_{i-1}^i \cdot \mathcal{V}_l = \{A_{i-1}^i \cdot \mathbf{r}_l^j \mid \mathbf{r}_l^j \in \mathcal{V}_l\}$ ;
14:      if  $\text{isPrismatic}(p_i)$  then
15:         $\mathcal{V}_l \leftarrow \text{sweepLin}(\mathcal{V}_l, p_{T_{s,i}}^-, p_{T_{s,i}}^+)$ ;
16:      else
17:         $\mathcal{V}_l \leftarrow \text{sweepRot}(\mathcal{V}_l, p_{T_{s,i}}^-, p_{T_{s,i}}^+)$ ;
18:      end if
19:    end if
20:  end for
21:   $\mathcal{SV} \leftarrow \mathcal{SV} \cup \{\mathcal{V}_l\}$ ;
22: end for

```

Algorithm 2. Swept Volume Calculation.

body parts, a radius parameter r is introduced and each convex swept volume \mathcal{V} is augmented by computing the Minkowski sum \oplus of its convex hull and a sphere of radius r :

$$\forall \mathcal{V}_i \in \mathcal{SV} \Rightarrow \mathcal{V}_i^* = \text{convhull}(\mathcal{V}_i) \oplus \{b \in \mathbb{R}^3: \|b\| \leq r\} \quad (41)$$

It is worth mentioning that the so-called “sphere-swept volumes” \mathcal{V}_i^* are used only for visualization purposes. As a matter of fact, according to the formulation of the safety constraints given in Section 2, it is possible to incorporate the radius r inside the clearance parameter d of equation (16) and to define the safety constraints directly for each swept volume \mathcal{V}_i .

4. Multiple depth-sensor measurements fusion

The occupancy prediction algorithm detailed in the previous Section requires human joint positions and velocities to be known in real-time. Depth sensors, like for instance Microsoft Kinect and ASUS Xtion, represent the most reasonable choice in order to track the motion of the full human upper body, since the 3-dimensional points acquired via skeletal tracking can be easily converted to human joint positions. However, not only these sensors do not provide any information regarding human joint velocities, but also numerical differentiation between consecutive joint positions is not a feasible option because of the noise affecting the measurements and also because of the interferences among the sensors.

Consequently, to estimate the kinematic configuration of a moving human worker using several depth sensors, we need to develop a sensor fusion strategy that:

- merges together the skeletal tracking measurements acquired by the different sensors;
- filters the noise affecting the acquired data;
- is able to estimate human joint velocities without relying on direct velocity measurements.

Given that both model (31) and the arm kinematics discretization introduced in Algorithm 1 are linear, a suitable solution to the sensor

fusion problem is represented by a Linear Kalman Filter (LKF) [42].

The main steps of the sensor fusion strategy are highlighted in Algorithm 3, where:

- $\mathbf{p}_{i,k}$ are the joint positions acquired at time step k by the i th sensor;
- $\pi_{i,k}$ are the kinematic parameters (i.e. the distances between the skeletal points) acquired at time step k by the i th sensor;
- $\text{valid}_{i,k}$ is a boolean variable whose value is *True* in case the measurements acquired at time step k by the i th sensor are valid and *False* otherwise;
- \mathbf{z}_k is the complete measurements vector at time step k ;
- $\hat{\mathbf{s}}_k = [\hat{\mathbf{p}}_k, \hat{\mathbf{p}}_k, \hat{\mathbf{p}}_k, \hat{\mathbf{p}}_k, \hat{\pi}_k]$ is the state estimate computed by the LKF at time step k . It contains estimated joint positions, velocities, accelerations, jerks, and it also includes the estimation of the distances between the skeletal points of the tracked individual;
- \mathbf{P}_k is the process covariance matrix of the LKF at time step k ;
- \mathbf{R}_k is the observation covariance matrix of the LKF at time step k ;

At first, all the acquired measurements are packed together (line 1). Then, the validity of the acquired measurements is checked (lines 2–4, details in Section 4.1.2). and the observation covariance matrix \mathbf{R}_k is updated (line 5), according to the procedure described in Section 4.1.1. At this point, an iteration of the LKF is executed (line 6). Finally, the resulting kinematic state estimate is refined to satisfy the bounds on joint positions (\mathbf{p}^{inf} and \mathbf{p}^{sup}), velocities ($\dot{\mathbf{p}}^{\text{inf}}$ and $\dot{\mathbf{p}}^{\text{sup}}$) and accelerations ($\ddot{\mathbf{p}}^{\text{inf}}$ and $\ddot{\mathbf{p}}^{\text{sup}}$) introduced by inequalities (18), (26), (34) and (35) (lines 7–9, details in Section 4.2). In the following, each phase of the algorithm is described in detail.

Differently from [34], our sensor fusion strategy is based on a single LKF that merges together all the measurements acquired by the different depth cameras. Furthermore, our approach works at Joint level, rather than in the Cartesian Space, thus allowing the resulting estimate to directly comply with the aforementioned limits on joint positions, velocities and accelerations.

4.1. KF-Based multiple depth sensor fusion

As stated before, in order to merge together the different

measurements acquired from the available depth sensors, we make use of a LKF. The adopted process model consists in a 3rd-order Taylor approximation for each joint variable. Regarding the walking kinematics, we here consider model (31) since it is linear and all the state variables are fully decoupled, while with regard to the arms kinematics we consider the model previously introduced in [40,43].

The filter state vector s contains joint positions, velocities, accelerations and jerks (see (43)). It also includes the kinematic parameters π that are estimated by imposing a constant dynamic, due to the fact that distances between the skeletal points can be considered constant.

In detail:

$$s_{k+1} = F s_k + \eta_k \quad (42)$$

$$s_k = \begin{bmatrix} p_k \\ \dot{p}_k \\ \ddot{p}_k \\ \pi_k \end{bmatrix} \quad F = \begin{bmatrix} I & \Delta t I & \frac{\Delta t^2}{2} I & \frac{\Delta t^3}{6} I & 0 \\ 0 & I & \Delta t I & \frac{\Delta t^2}{2} I & 0 \\ 0 & 0 & I & \Delta t I & 0 \\ 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & I \end{bmatrix} \quad (43)$$

In particular, $\eta_k \sim \mathcal{N}(0, G)$ models the process noise, whose covariance matrix G can be parametrized as follows. For each block of the state vector we consider the Cauchy remainder of the Taylor approximations contained in matrix F :

$$G = \begin{bmatrix} \sigma_p^2 \frac{\Delta t^4}{24} I & 0 & 0 & 0 & 0 \\ 0 & \sigma_{\dot{p}}^2 \frac{\Delta t^3}{6} I & 0 & 0 & 0 \\ 0 & 0 & \sigma_{\ddot{p}}^2 \frac{\Delta t^2}{2} I & 0 & 0 \\ 0 & 0 & 0 & \sigma_{\pi}^2 \Delta t I & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (44)$$

where standard deviations σ_p , $\sigma_{\dot{p}}$, $\sigma_{\ddot{p}}$ and σ_{π} depend on the entity of the first neglected derivative (jerk).

Regarding the observation model, since the inverse kinematics of the human motion model can be computed in closed-form, we consider as observed output vector z_k already introduced in Algorithm 3.

Consequently we consider as observation model the following linear transformation:

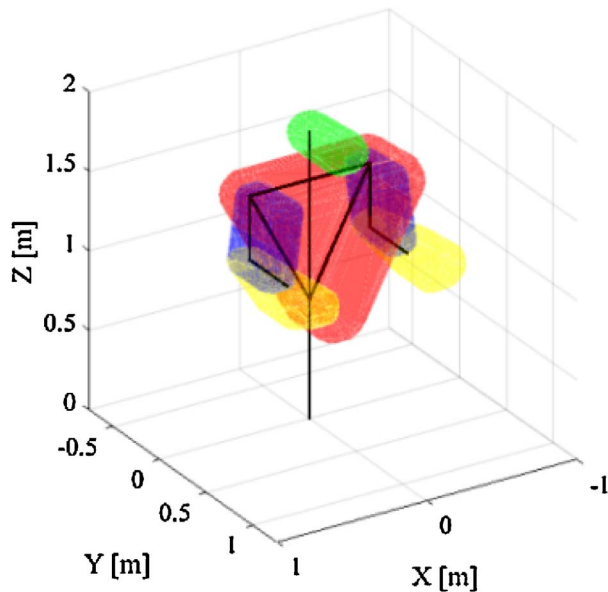


Fig. 9. Swept volumes computed on the basis of joint positions and velocities and fixed bounds on joint accelerations.

$$z_k = H s_k + \zeta_k \quad (45)$$

$$H = \begin{bmatrix} I & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & I \end{bmatrix} \quad (46)$$

Moreover, $\zeta_k \sim \mathcal{N}(0, R_0)$ models the measurement noise, whose covariance matrix is given by:

$$R_0 = \sigma_z^2 I \quad (47)$$

where standard deviation σ_z depends on the accuracy measure that the OpenNI driver² returns for each skeletal points acquired.

4.1.1. Synchronizing the LKF and the data acquisition process

A relevant issue in the design of this sensor fusion strategy is the lack of synchronization between the acquisition process and the process implementing the LKF. As a matter of fact, in order to ensure accurate estimation we need to run the LKF at a frequency that usually is much greater than the data acquisition frequency. Consequently the filter will execute several iterations on the basis of the same set of measurements.

A possible solution consists in updating the observation covariance matrix R_k according to the presence (or not) of new measurements, in such a way that each element on the main diagonal of R_k follows a saw tooth shape. More specifically, every time the LKF process receives a new set of measurements, the corresponding blocks of matrix R_k are reset to $\sigma_z^2 I$. On the other hand, whenever the same set of measurements is re-used, the corresponding blocks in R_k are updated by adding $\sigma_z^2 I$. In this way we obtain an uncertainty that is reset to a starting value whenever a new measurement arrives and that grows linearly in time between two consecutive measurements.

4.1.2. A-posteriori state estimation in presence of non-valid measurements

Data acquired from depth sensors can suffer from non-valid measurements due to various reasons: occlusions, distance from the sensor, field of view, etc. To this regard, the OpenNI driver outputs a confidence measure associated to each skeletal point, ranging from 0 to 1. For instance, when a skeletal point is occluded by another object, its confidence rapidly decreases to zero. The same situation happens whenever the point leaves the field of view of the camera or when it is too near with respect to the sensor. As far as the possibility of objects being merged with the human skeleton is concerned, the Kalman Filter is naturally able to diminish the impact of such outliers. Moreover, evident outliers can be identified by comparing the acquired skeletal points with the output of the human forward kinematics applied to the KF state estimate.

In Algorithm 3, for each set of measurements $p_{i,k}$ and $\pi_{i,k}$ a boolean variable, named $valid_i$, is instantiated. Whenever the product (or sum) of the confidence measures is higher than a tunable threshold, $valid_i$ is set to true, otherwise it is set to false. In case at time step k there are no valid measurements available, the filter directly outputs the a-priori estimate:

$$\hat{s}_{k|k-1} = F \hat{s}_{k-1} \quad (48)$$

$$P_{k|k-1} = F P_{k-1} F^T + G \quad (49)$$

$$\hat{s}_k \leftarrow \hat{s}_{k|k-1} \quad (50)$$

$$P_k \leftarrow P_{k|k-1} \quad (51)$$

while if there is at least one valid measurement, the filter also executes the prediction update and outputs the a-posteriori state estimation:

² OpenNI - The standard framework for 3D sensing, <http://www.openni.org/>, March 2013

```

1:  $\mathbf{z}_k = [\mathbf{p}_{1,k}, \dots, \mathbf{p}_{n,k}, \boldsymbol{\pi}_{1,k}, \dots, \boldsymbol{\pi}_{n,k}]$ ;
2: for all  $i \in [1, \dots, n]$  do
3:    $\text{valid}_{i,k} \leftarrow ()$ ;
4:    $\text{valid}_{i,k} \leftarrow \text{valid}(\mathbf{p}_{i,k}, \boldsymbol{\pi}_{i,k})$ ;
5: end for
6:  $\mathbf{R}_k \leftarrow \text{updateObservationCov}()$ ;
7:  $[\hat{\mathbf{s}}_k, \mathbf{P}_k] \leftarrow \text{LKF}(\mathbf{z}_k, \text{valid}_{1,k}, \dots, \text{valid}_{n,k} \mathbf{R}_k)$ ;
8:  $\hat{\mathbf{p}}_k \leftarrow \text{saturateJointPos}(\hat{\mathbf{p}}_k, \mathbf{p}^{inf}, \mathbf{p}^{sup})$ ;
9:  $\hat{\dot{\mathbf{p}}}_k \leftarrow \text{saturateJointVel}(\hat{\mathbf{p}}_k, \dot{\mathbf{p}}_k, \dot{\mathbf{p}}^{inf}, \dot{\mathbf{p}}^{sup})$ ;
10:  $\hat{\ddot{\mathbf{p}}}_k \leftarrow \text{saturateJointAcc}(\hat{\mathbf{p}}_k, \hat{\dot{\mathbf{p}}}_k, \hat{\ddot{\mathbf{p}}}_k, \ddot{\mathbf{p}}^{inf}, \ddot{\mathbf{p}}^{sup})$ ;

```

Algorithm 3. Sensor Fusion Algorithm.

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - H\hat{\mathbf{s}}_{k|k-1} \quad (52)$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} H^T (H \mathbf{P}_{k|k-1} H^T + \mathbf{R}_k)^{-1} \quad (53)$$

$$\hat{\mathbf{s}}_{k|k} = \hat{\mathbf{s}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k \quad (54)$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - H\mathbf{K}_k) \mathbf{P}_{k|k-1} \quad (55)$$

$$\hat{\mathbf{s}}_k \leftarrow \hat{\mathbf{s}}_{k|k} \quad (56)$$

$$\mathbf{P}_k \leftarrow \mathbf{P}_{k|k} \quad (57)$$

In order to prevent non-valid measurements from affecting the a-posteriori state estimate $\hat{\mathbf{s}}_{k|k}$, we set to zero all the corresponding components of the innovation signal $\tilde{\mathbf{y}}_k$ in equation (54):

$$\forall i \in [1, n], (\text{valid}_i = \text{False}) \Rightarrow \tilde{\mathbf{y}}_{i,k} \leftarrow 0 \quad (58)$$

As a matter of fact this solution is equivalent to consider a time-varying observation matrix H_k , whose blocks switch between the identity and the null matrix. A similar solution has been proposed in [44] where, in case at time step k an observation is absent, the corresponding element of the diagonal noise covariance matrix is set to $\lim_{\sigma \rightarrow +\infty} \sigma^2$. Consequently, the corresponding row of matrix \mathbf{K}_k tends to the null vector, hence trivializing the update formulas (53)–(55). However, our approach differs from this solution in several ways. First, we treat in the same way missing and non valid measurements, thus making the resulting estimation more robust with respect to sensor occlusions and failures. Furthermore, as explained in Section 4.1.1, our solution exploits a time-varying noise observation matrix \mathbf{R}_k in order to address also unsynchronized observations.

Finally, it is worth noticing that the presence of the time-varying matrices H_k and \mathbf{R}_k does not invalidate the stability and correctness property of the LKF, since the corresponding linearity and Gaussianity requirements are always met.

4.2. Bounded kinematic state estimation

Clearly, the estimation computed by the LKF $\hat{\mathbf{s}}_k$ does not necessarily satisfy the chosen bounds on joint positions (\mathbf{p}^{inf} and \mathbf{p}^{sup}), velocities ($\dot{\mathbf{p}}^{inf}$ and $\dot{\mathbf{p}}^{sup}$) and accelerations ($\ddot{\mathbf{p}}^{inf}$ and $\ddot{\mathbf{p}}^{sup}$). Consequently, it might be necessary to refine the kinematic state estimation in order to enforce its coherence with respect to the aforementioned bounds. To this purpose we consider the following QP problem:

$$\min_{\tilde{\mathbf{s}}} \|\tilde{\mathbf{s}} - \hat{\mathbf{s}}\|^2 \quad (59a)$$

$$\mathbf{A}\tilde{\mathbf{s}} \leq \mathbf{b} \quad (59b)$$

where:

$$\mathbf{A} = \begin{bmatrix} -\mathbf{I} \\ \mathbf{I} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} -\mathbf{s}^{inf} \\ +\mathbf{s}^{sup} \end{bmatrix} \quad (60)$$

Thanks to the theoretical results presented in [45,46], problem (59) can be solved by finding the solution of an equivalent QP problem. This problem can be formulated by considering the same cost metric, but taking into account only the active set of constraints, expressed as equality constraints. As a matter of fact, this QP problem can be solved by simply imposing, to each element of the original estimate $\hat{\mathbf{s}}$, the corresponding saturation functions:

$$\forall \tilde{s}_i \in \tilde{\mathbf{s}} \Rightarrow \tilde{s}_i = \min(\max(\hat{s}_i, s_i^{inf}), s_i^{sup}) \quad (61)$$

As a consequence, the estimation computed by the LKF can be easily modified in order to satisfy the imposed bounds. At first, we update the estimated joint positions according to the imposed bounds:

$$\hat{\mathbf{p}}_k \leftarrow \tilde{\mathbf{p}}_k = \max(\mathbf{p}^{inf}, \min(\hat{\mathbf{p}}_k, \mathbf{p}^{sup})) \quad (62)$$

where the $\max(\cdot)$ and the $\min(\cdot)$ functions are applied element-wise. Then, the refined joint positions $\hat{\mathbf{p}}_k$ are used in combination with the imposed velocity bounds to update the estimated joint velocities, thus obtaining:

$$\dot{\tilde{\mathbf{p}}}_k^{inf} = \max(\dot{\mathbf{p}}_k^{inf}, (\mathbf{p}_k^{inf} - \hat{\mathbf{p}}_k)/\Delta t) \quad (63)$$

$$\dot{\tilde{\mathbf{p}}}_k^{sup} = \min(\dot{\mathbf{p}}_k^{sup}, (\mathbf{p}_k^{sup} - \hat{\mathbf{p}}_k)/\Delta t) \quad (64)$$

$$\hat{\dot{\mathbf{p}}}_k \leftarrow \tilde{\dot{\mathbf{p}}}_k = \max(\dot{\tilde{\mathbf{p}}}_k^{inf}, \min(\dot{\tilde{\mathbf{p}}}_k^{sup})) \quad (65)$$

Finally, by applying this update procedure to joint accelerations, we obtain the estimation of the human worker's kinematic state that best fits the acquired measurements, while being fully coherent with respect to position, velocity and acceleration bounds. Clearly, the obtained kinematic state is sent back to the LKF in order to keep the evolution of the filter coherent with respect to the output of the sensor fusion algorithm.

5. Safety-aware trajectory generation algorithm

In this Section the safety-oriented trajectory generation algorithm is discussed. The algorithm is formulated in terms of a QP problem whose solution, at each time step k , consists in a set of joint reference accelerations, treated as control variables: $\mathbf{u}_k = \ddot{\mathbf{q}}_k^{ref}$.

Several control strategies based on constrained optimization have been proposed in the last few years. For instance, in [47,48] a constraint-based strategy is proposed to allow specification of complex tasks. More specifically, QP techniques have been used in [49,50] to address also the problem of kinematic redundancy, while a constraint-based algorithm working at the acceleration level can be found in [51]. Finally, several constraint-based control strategies have been proposed to manage a prioritized hierarchy of tasks, like for instance [52–54]. In this context, the main distinctive feature of the algorithm here proposed is represented by the fact that it incorporates the previously defined safety constraints in order to reactively adapt the pre-programmed

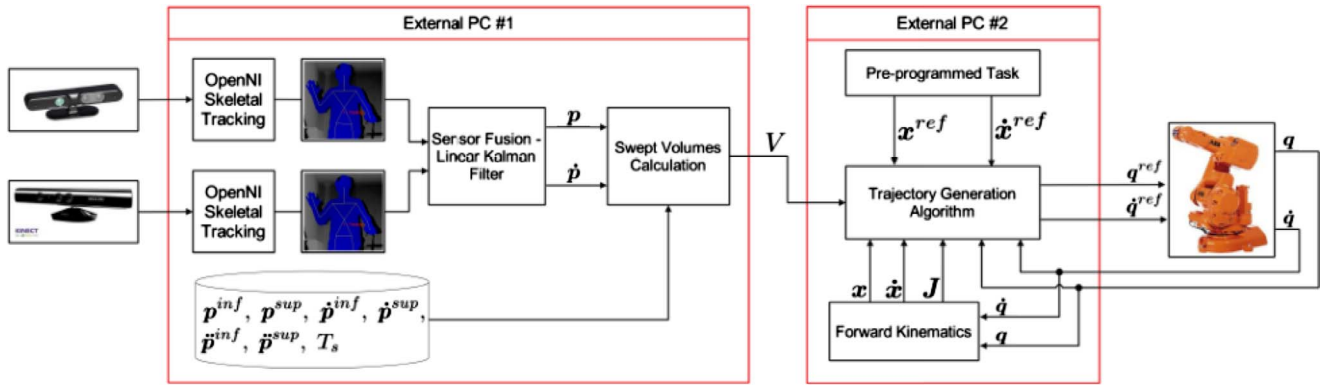


Fig. 10. Experimental setup - deployment diagram.

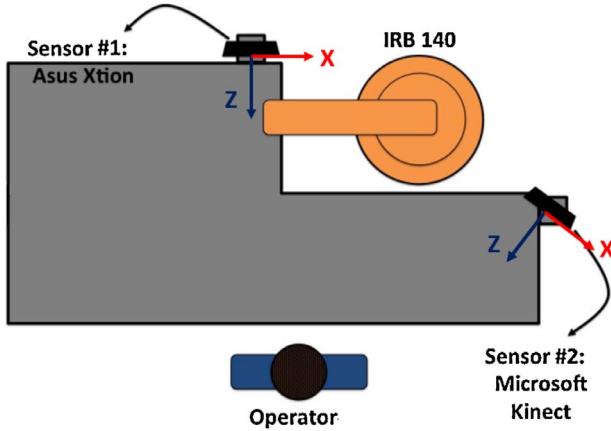


Fig. 11. Experimental setup - position of robot, human user and depth sensors.

path, thus achieving a fruitful trade-off between operators' safety and machine productivity.

In detail, at each time step k the QP problem is solved and the obtained accelerations are used to update the next step position and velocity references (\mathbf{q}_{k+1}^{ref} and $\dot{\mathbf{q}}_{k+1}^{ref}$, respectively) that will be actuated. The QP problem is formulated as a tracking problem so that the manipulator tries to follow the pre-programmed trajectory as much as possible while satisfying the safety constraints computed considering each human swept volume as an obstacle. Whenever exact tracking is not feasible, the algorithm selects the acceleration values that keep the manipulator as close as possible with respect to the pre-programmed trajectory, among the feasible ones.

Even though there exist in the scientific literature some contributions [55,56] that combine potential field-based repulsive velocity terms [5] with safety constraints, we did not adopt this strategy for two main reasons. First, repulsive velocities can cause excessive disruption of the pre-programmed trajectory, thus decreasing the productivity of the robot. Secondly, even an evasive motion, if sufficiently fast and sudden, can undermine the level of safety perceived by the human worker.

More in detail, the complete QP problem is reported in (66).

$$\min_{\mathbf{u}_k} \left(\|\dot{\mathbf{x}}_{k+1} - \dot{\mathbf{x}}_{k+1}^{ref}\|_{Q_v}^2 + \|\mathbf{x}_{k+1} - \mathbf{x}_{k+1}^{ref}\|_{Q_p}^2 \right) \quad (66a)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \mathbf{J}_k \dot{\mathbf{q}}_k + \frac{\Delta t^2}{2} (\ddot{\mathbf{J}}_k \dot{\mathbf{q}}_k + \mathbf{J}_k \mathbf{u}_k) \quad (66b)$$

$$\dot{\mathbf{x}}_{k+1} = \mathbf{J}_k \dot{\mathbf{q}}_k + \Delta t (\ddot{\mathbf{J}}_k \dot{\mathbf{q}}_k + \mathbf{J}_k \mathbf{u}_k) \quad (66c)$$

$$\mathbf{q}^{inf} \leq \mathbf{q}_k + \Delta t \dot{\mathbf{q}}_k + \frac{1}{2} \Delta t^2 \mathbf{u}_k \leq \mathbf{q}^{sup} \quad (66d)$$

Table 2

Values of radius parameters involved in the computation of the sphere-swept volumes.

Body Part	Radius [m]
Head	0.12
Thorax	0.15
Upper Arm	0.07
Lower Arm	0.07

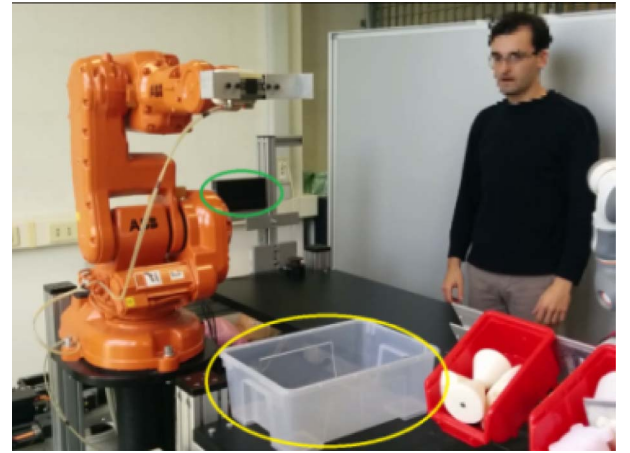


Fig. 12. Experimental validation scenario: the manipulator has to pick the metal pieces (highlighted in green) from the rack and deposit them into the plastic box (highlighted in yellow) under the supervision of a human worker. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$\dot{\mathbf{q}}^{inf} \leq \dot{\mathbf{q}}_k + \Delta t \mathbf{u}_k \leq \dot{\mathbf{q}}^{sup} \quad (66e)$$

$$\mathbf{u}^{inf} \leq \mathbf{u}_k \leq \mathbf{u}^{sup} \quad (66f)$$

$$-\dot{\mathbf{x}}^{max} \leq \dot{\mathbf{x}}_{k+1} \leq \dot{\mathbf{x}}^{max} \quad (66g)$$

$$-\dot{\mathbf{x}}^{max} \leq \ddot{\mathbf{J}}_k \dot{\mathbf{q}}_k + \mathbf{J}_k \mathbf{u}_k \leq \dot{\mathbf{x}}^{max} \quad (66h)$$

$$\forall \mathcal{V}_i \in \mathcal{V}, \forall \mathbf{r}_i^j \in \mathcal{V}_i \Rightarrow \mathbf{E}_k(\mathbf{r}_i^j)(\dot{\mathbf{q}}_k + \Delta t \mathbf{u}_k) \leq \mathbf{f}_k(\mathbf{r}_i^j) \quad (66i)$$

where:

- \mathbf{x}_k^{ref} and $\dot{\mathbf{x}}_k^{ref}$ are the pre-programmed robot task-space position and velocity at time step k ;
- \mathbf{x}_k and $\dot{\mathbf{x}}_k$ are the actual robot task-space position and velocity at time step k ;
- \mathbf{Q}_p and \mathbf{Q}_v are diagonal matrices acting as weights for the task-space position and velocity error, respectively, inside the tracking cost function;

Table 1

Position, velocity and acceleration bounds for each joint of human kinematic chain (37).

Joint	Min Pos.	Max Pos.	Min Vel.	Max Vel.	Min Acc.	Max Acc.
x	$-\infty$ m	$+\infty$ m	0 m/s	$+0.80$ m/s	-0.10 m/s ²	$+0.10$ m/s ²
y	$-\infty$ m	$+\infty$ m	-0.80 m/s	$+0.80$ m/s	-0.10 m/s ²	$+0.10$ m/s ²
θ	$-\pi$ [rad]	$+\pi$ [rad]	$-\pi/4$ [rad/s]	$+\pi/4$ [rad/s]	$-\pi/8$ [rad/s ²]	$+\pi/8$ [rad/s ²]
ρ	$-\pi/6$ [rad]	$+\pi/2$ [rad]	$-\pi/10$ [rad/s]	$+\pi/10$ [rad/s]	$-\pi/20$ [rad/s ²]	$+\pi/20$ [rad/s ²]
$\alpha_{r,1}$	$-\pi/20$ [rad]	$+8\pi/9$ [rad]	$-\pi/2$ [rad/s]	$+\pi/2$ [rad/s]	$-\pi/4$ [rad/s ²]	$+\pi/4$ [rad/s ²]
$\alpha_{r,2}$	inequality (34b)		$-\pi/2$ [rad/s]	$+\pi/2$ [rad/s]	$-\pi/4$ [rad/s ²]	$+\pi/4$ [rad/s ²]
$\alpha_{r,3}$	inequality (34c)		$-\pi/2$ [rad/s]	$+\pi/2$ [rad/s]	$-\pi/4$ [rad/s ²]	$+\pi/4$ [rad/s ²]
$\alpha_{r,4}$	$+\pi/9$ [rad]	$+\pi$ [rad]	$-3\pi/10$ [rad/s]	$+3\pi/10$ [rad/s]	$-3\pi/20$ [rad/s ²]	$+3\pi/20$ [rad/s ²]
$\alpha_{l,1}$	$-\pi/20$ [rad]	$+8\pi/9$ [rad]	$-\pi/2$ [rad/s]	$+\pi/2$ [rad/s]	$-\pi/4$ [rad/s ²]	$+\pi/4$ [rad/s ²]
$\alpha_{l,2}$	inequality (34b)		$-\pi/2$ [rad/s]	$+\pi/2$ [rad/s]	$-\pi/4$ [rad/s ²]	$+\pi/4$ [rad/s ²]
$\alpha_{l,3}$	inequality (34c)		$-\pi/2$ [rad/s]	$+\pi/2$ [rad/s]	$-\pi/4$ [rad/s ²]	$+\pi/4$ [rad/s ²]
$\alpha_{l,4}$	$+\pi/9$ [rad]	$+\pi$ [rad]	$-3\pi/10$ [rad/s]	$+3\pi/10$ [rad/s]	$-3\pi/20$ [rad/s ²]	$+3\pi/20$ [rad/s ²]

- \mathbf{q}_k and $\dot{\mathbf{q}}_k$ are the actual robot joint-space position and velocity, respectively, at time step k ;
- \mathbf{J}_k stands for $\mathbf{J}(\mathbf{q}_k)$;
- Δt is the robot controller sample time.

The cost function (66a) is meant to guarantee best tracking performance of the pre-programmed trajectory in terms of displacement and velocity. Constraints in (66b) and (66c) represent a second order Taylor approximation of the forward kinematics of the manipulator, while those in (66d), (66e) and (66f) implement joint position, velocity and acceleration limits, respectively, as well as (66h) and (66g) represent Cartesian acceleration and velocity limits.

In case QP problem (66) does not allow a solution, the control vector \mathbf{u}_k is computed in order to bring joint velocities to zero in the minimum time:

$$\forall u_{k,i} \in \mathbf{u}_k \Rightarrow u_{k,i} = \min \left(u_i^{\sup}, \max \left(\frac{-\dot{q}_{k,i}}{\Delta t}, u_i^{\inf} \right) \right) \quad (67)$$

Please notice that this procedure can be considered as a last resort. As a matter of fact, the formulation of the safety constraints already takes into account the prediction of the space that the human worker can occupy within the robot's worst case braking time. Consequently, the violation of the safety constraints is prevented, since the robot is always given enough time to either deviate from the pre-programmed path or to slow down and eventually stop. Clearly, in a real-world scenario, phenomena such as measurement noise can alter the estimated human kinematic configuration, thus making it necessary to implement this halting strategy.

Finally, position and velocity references at the next time step are obtained as follows:

$$\dot{\mathbf{q}}_{k+1}^{\text{ref}} = \dot{\mathbf{q}}_k^{\text{ref}} + \Delta t \mathbf{u}_k \quad (68)$$

$$\mathbf{q}_{k+1}^{\text{ref}} = \mathbf{q}_k^{\text{ref}} + \Delta t \dot{\mathbf{q}}_k^{\text{ref}} + \frac{\Delta t^2}{2} \mathbf{u}_k \quad (69)$$

and sent to the lower level axis controller. For the sake of completeness, it is worth mentioning that The QP problem is solved using qpOASES solver [57] that relies on the Online Active-Set Strategy [58].

6. An experimental case study

In this Section we describe a relevant case study and we present the result obtained by applying our proposed approach. The robot has to accomplish a simple pick and place task while cooperating with the human worker.

6.1. Experimental setup

For the implementation and validation of the proposed trajectory generation algorithm, we here consider the experimental setup sketched in Fig. 10. It consists of:

- ABB IRB 140 robot: a 6 axes industrial robot with 6 kg maximum payload. Its position is controlled by an industrial ABB IRC 5 controller connected to an External PC through an Ethernet-based interface;
- Microsoft Kinect and ASUS Xtion: two RGB-D motion sensing cameras equipped with the OpenNI drivers used to detect the presence of a human worker inside the robotic cell and to perform skeletal tracking. Both cameras are calibrated with respect to the robot base using the ICP algorithm [59]. In order to reduce interferences, sensors are positioned in such a way that the Z axis of the corresponding Cartesian frames are not parallel (see Fig. 11);
- External PC #1: a real-time PC that takes care of running the sensor fusion algorithm according to the measurements acquired from the two depth sensors (see Section 4) and to compute the human swept volumes on the basis of the estimated worker's kinematic configuration with respect to the model described in Section 3. This PC also stores the parameters needed to compute the swept volumes: position, velocity and acceleration upper and lower bounds along with the braking time T_s ;
- External PC #2: a real-time PC that runs the trajectory generation algorithm on the basis of both the kinematic configuration of the robot and the predicted occupancy of the human worker. It is interfaced to External PC #1 with a standard Ethernet interface and it is connected to the IRC5 controller through a real-time Ethernet-based interface (see [60] for further details). It runs under Linux OS with the Xenomai patch, that makes it a hard real-time system. Using the real-time Ethernet-based interface, it is possible to develop and execute a control algorithm, named "External Controller", that communicates in real-time with the IRC5 controller with a frequency of 250 Hz. Full-duplex communication allows the External Controller to acquire data regarding the kinematic configuration of the manipulator and to override the joint position and velocity reference signals the IRC5 sends to the low-level joint controllers.

It is worth mentioning that using OpenMP pragma directives [61] it is possible to run the modules implemented on External PC #1 in real-time (at a frequency of 205 Hz). On the other hand, as far as the implementation of the trajectory generation algorithm is concerned, pre-processing techniques [62] are employed in order to reduce the number of constraints that are actually considered during the solution of QP problem (66), thus ensuring that also the QP solver provides a result

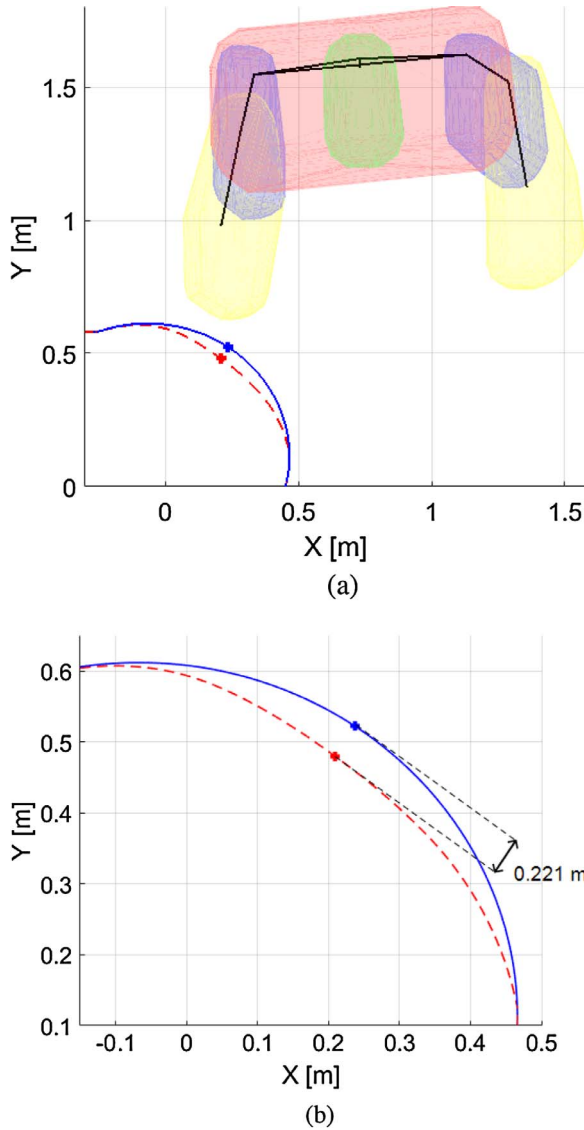


Fig. 13. Nominal end-effector trajectory (blue solid line) versus trajectory followed in presence of the human worker (red dashed line). (a): pick-and-place iteration in presence of a human operator. (b): detail of deviation between the actual trajectory and the nominal one. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

within a single time step (4 ms).

Finally, Table 1 lists the values of the parameters involved in the calculation of the reachable sets, while Table 2 displays the values of the different radii used to determine the sphere-swept volumes (see equation (41)). These worst-case bounds have been chosen in order to obtain a prediction of human occupancy that is reasonably conservative and, at the same time, that does not result in excessively large swept volumes. Beside these bounds, other relevant parameters values are:

$$T_s = 0.377 \text{ s} \quad (\text{see [24] for details}) \quad (70)$$

$$Q_p = 0.10 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (71)$$

$$Q_v = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (72)$$

6.2. Experimental validation

The trajectory generation algorithm here proposed has been validated considering a typical pick-and-place scenario. The robot picks some metal pieces from a rack and leaves them inside a plastic box under the supervision of a human worker standing inside its workspace, as depicted in Fig. 12.

Fig. 13(a) and (b) show a comparison between two distinct end-effector Cartesian trajectories: the nominal one (blue solid line) and the one followed by the robot in presence of the human worker (red dashed line). More in depth, Fig. 13(a) not only shows the difference between the two trajectories, but it also highlights two points. The blue one belongs to the nominal trajectory, while the red one is the corresponding point on the actual trajectory determined on the basis of the human swept volumes also pictured in the Figure. A closer look to the deviation can be taken in Fig. 13(b), where the Euclidean distance between the two points 0.221 m is also reported. As a matter of fact, the experimental data clearly demonstrate that our algorithm allows the manipulator to successfully complete the task, in terms of reaching the endpoints of the nominal trajectory, even though the presence of a moving human worker forces the robot to deviate from the pre-programmed path.

A similar experiment is documented by the series of screenshots shown in Figs. 15 and 16, where it can be seen how the proposed algorithm adapts the pre-programmed trajectory in order to preserve the human worker's safety, while allowing the robot to successfully complete the task. Moreover, the attached video file integrates the experiment description and it also provides a graphical reconstruction of the swept volumes computed during the experiment.

On the other hand, Fig. 14 shows that, in order to satisfy the safety constraints, the control algorithm not only enforces deviations from the pre-programmed path, but also adapts the manipulator speed. More precisely, whenever the robot gets too close to the human worker, its speed is reduced in order to satisfy the safety constraints. It is worth noticing that the combination of evasive motion and speed reduction allows the robot to execute the task without necessarily relying on a full stop in order to ensure the human worker's safety.

7. Discussion and comparison with previous proposed approaches

This section addresses the advantages of the proposed trajectory generation algorithm compared to the solutions presented in our

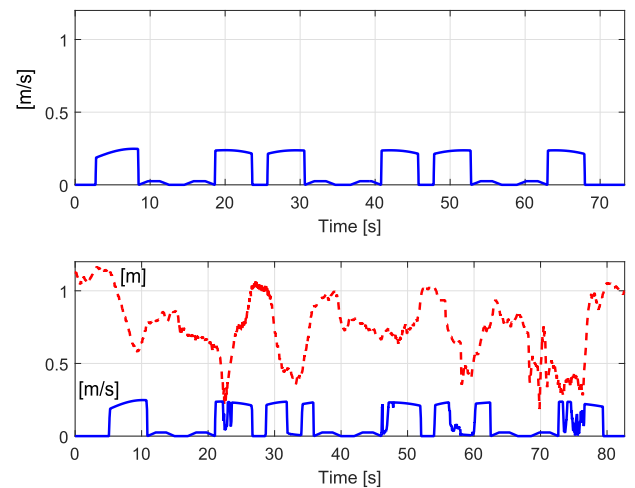


Fig. 14. Top plot: nominal end-effector Cartesian velocity (blue solid line). Bottom plot: end-effector Cartesian velocity in presence of the human worker (blue solid line) versus the minimum human-robot distance (red dashed line). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

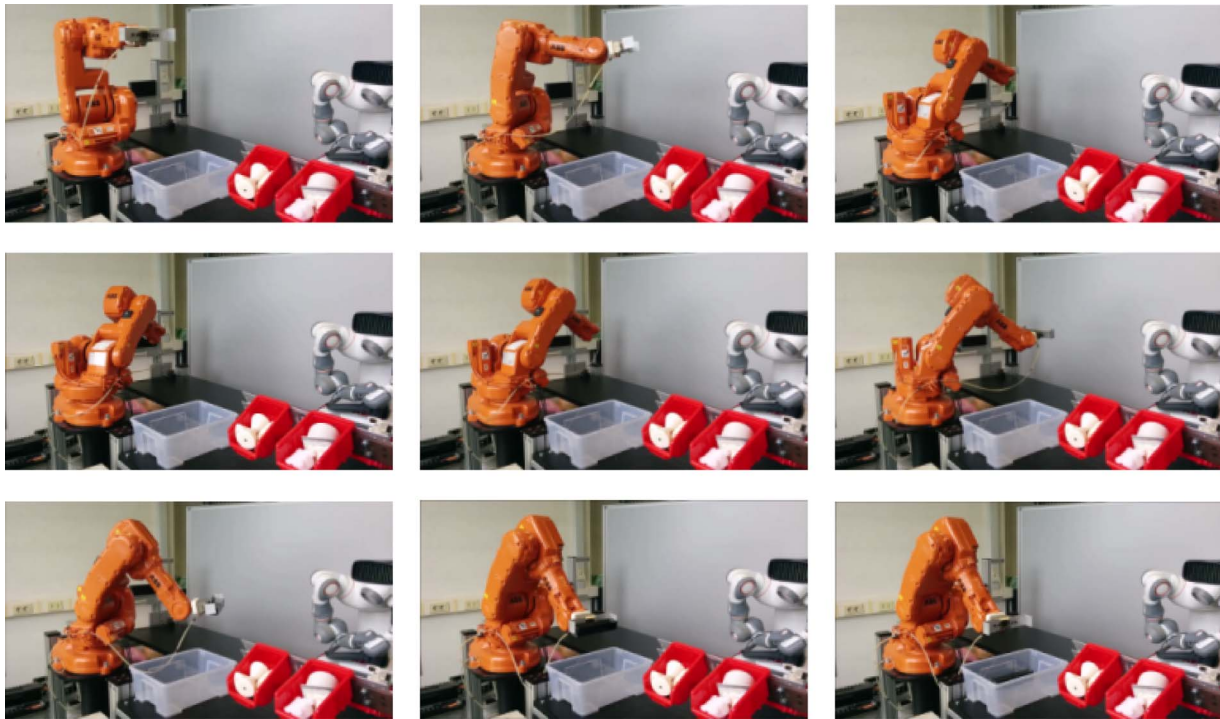


Fig. 15. Screenshots taken during a pick-and-place iteration without supervising human worker. The robot follows the pre-programmed trajectory.

previous works [24–26]. As regards the prediction of human occupancy, in [26] the calculation of the swept volumes was merely based on a single set of human joint positions acquired via skeletal tracking from a single depth sensor.

The most significant improvement introduced by the algorithm here proposed is represented by the fact that the new prediction of human occupancy is considerably less conservative with respect to the previous one. As a matter of fact, in [26] reachable sets are computed on the

basis of estimated joint positions and maximum/minimum joint velocities, while the procedure here proposed relies on the estimation of current joint positions and velocities, introducing fixed upper and lower bounds only at the joint acceleration level. The result is exemplified in Fig. 17 where, given a trajectory tracked by the multiple camera system, Swept Volumes computed according to the two alternatives approaches are compared. Clearly, not only the new approach produces smaller and more accurate swept volumes, but it also entails

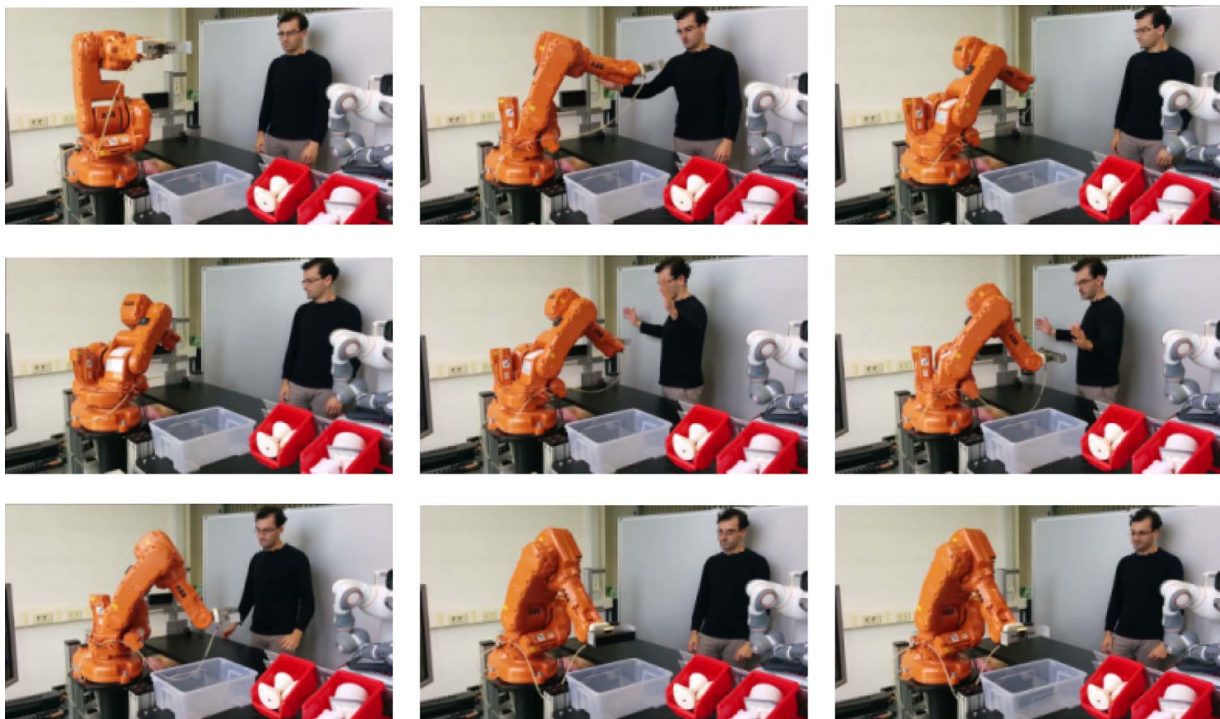


Fig. 16. Screenshots taken during a pick-and-place iteration in presence of a supervising human worker. The algorithm adapts the trajectory followed by the manipulator in order to ensure the worker's safety.

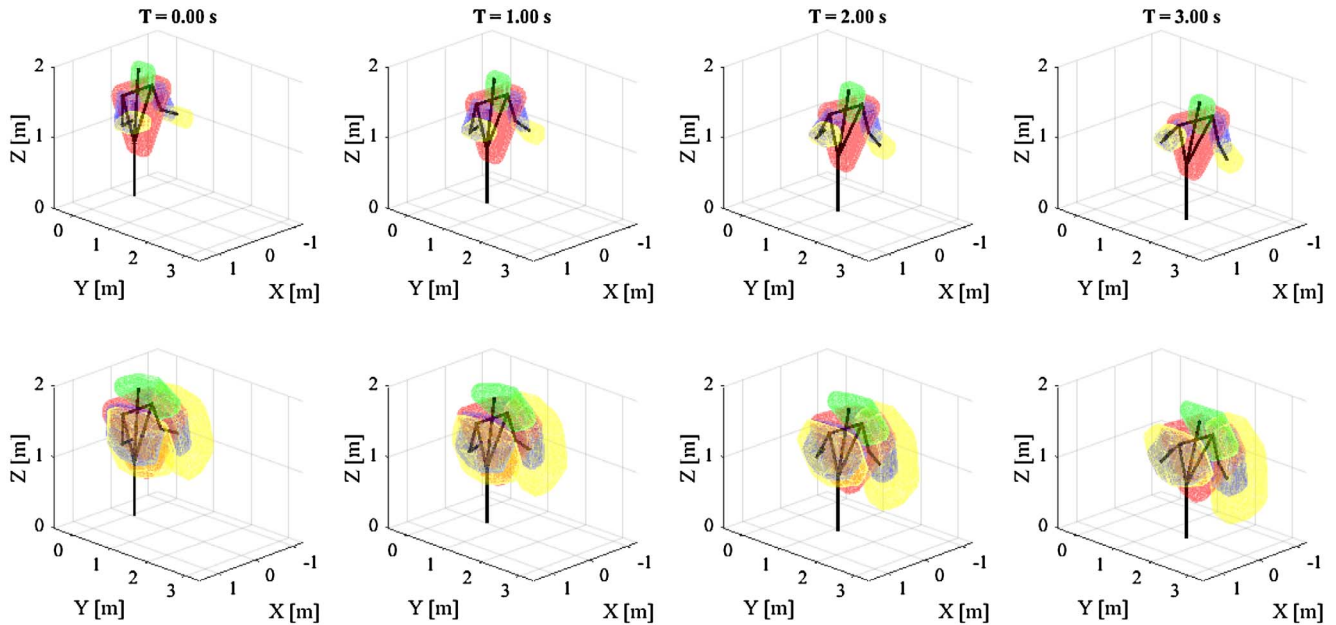


Fig. 17. Comparison between swept volumes algorithms. From left to right, Swept Volumes computed at $T = 0\text{ s}$, $T = 1\text{ s}$, $T = 2\text{ s}$ and $T = 3\text{ s}$. Top row: Swept Volumes computed according to current algorithm (joint positions and velocity). Bottom row: Swept Volumes computed according to [26] (only joint positions).

significantly smaller, yet conservative, safety constraints.

Another significant improvement is represented by the introduction of the sensor fusion strategy. As a matter of fact, the possibility to merge together the measurements acquired by multiple sensors allows to obtain a much more accurate and noise-free estimation of human joint positions (as shown in Fig. 18). Moreover, the sensor fusion algorithm is able to estimate human joint velocities and to provide robustness with respect to occlusions, invalid measures and sensor failures.

Regarding the safety-oriented trajectory generation strategy, the approach here proposed represents a significant improvement with respect to the kinematic scaling algorithm presented in [24–26]. As a matter of fact, the previously proposed solution was able to enforce the safety constraints by slowing down or completely stopping the robot whenever a human worker reached its proximities. If we consider a task that requires the geometry of the pre-programmed path to be preserved and if the robot is equipped with a closed-ended controller (i.e. a controller that does not provide the possibility to override joint position

and velocity references in real-time), this solution is perfectly suitable. As a matter of fact, traditional robot programming languages usually offer the possibility to scale the maximum admissible speed of the manipulator.

On the other hand, in case the robot controller is open (see Section 6.1) and if the geometry of the pre-programmed trajectory can be modified, it might be possible to satisfy the safety constraints not only by reducing the robot speed, but also by deviating from the pre-programmed path. In these scenarios, the approach here proposed can remarkably reduce the amount of time during which the manipulator is still, thus maximizing efficiency, while respecting at the same time safety requirements. This claim can be clearly demonstrated by comparing the experimental results presented in [26] with the ones here discussed. As a matter of fact, whenever the minimum distance between the robot and the Swept Volumes is equal to 0.50 m or less, the kinematic scaling algorithm outputs a null robot velocity in 86.44% of the cases, while the algorithm here proposed stops the robot only in 30.06% of the cases. Clearly, in order to avoid excessive task disruption, the trajectory generation algorithms tries to minimize this deviation in order to preserve the pre-programmed trajectory as much as possible.

8. Conclusions

This paper presents a trajectory generation algorithm for safe HRC. The proposed solution relies on multiple depth sensors for real-time tracking of human motion and on a sensor fusion strategy that merges the different measurements in order to estimate the kinematic configuration of the human worker. On the basis of this estimation, and given a time limit, it is possible to predict the future space occupancy of the human worker in terms of swept volumes. The trajectory generation algorithm takes into account these volumes as obstacles and modifies the robot trajectory in order to ensure the human worker's safety, while performing the pre-programmed task. Details on the actual implementation and deployment of the proposed approach are discussed and results of validation experiments conducted on a ABB IRB140 robot are presented.

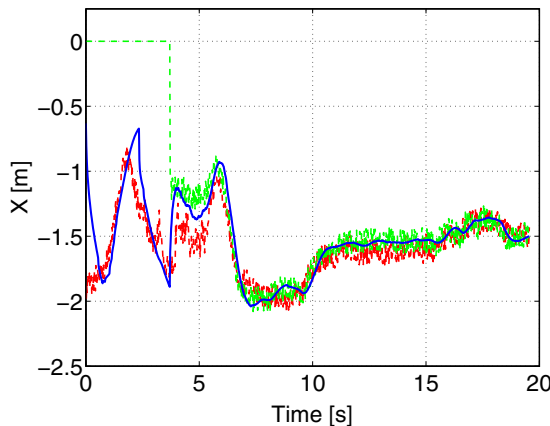


Fig. 18. Comparison between x coordinate of kinematic configuration vector (37) values acquired by ASUS Xtion (dashed red line), acquired from MS Kinect (dashed green line - equal to zeros until the sensor starts tracking the human worker) and estimated by sensor fusion algorithm (solid blue line). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

References

- [1] ANSI/RIA R15.06. Safety requirements for industrial robots and robot systems. 1999.
- [2] Haddadin S, Albu-Schäffer A, Hirzinger G. Requirements for safe robots: measurements, analysis and new insights. *Int J Rob Res* 2009;28(11–12):1507–27.
- [3] Haddadin S, Haddadin S, Khoury A, Rokahr T, Parusel S, Burgkart R, et al. On making robots understand safety: embedding injury knowledge into control. *Int J Rob Res* 2012;31(13):1578–602.
- [4] Zanchettin AM, Lacevic B, Rocco P. Passivity-based control of robotic manipulators for safe cooperation with humans. *Int J Control* 2015;88(2):429–39.
- [5] Khatib O. Real-time obstacle avoidance for manipulators and mobile robots. *Int J Rob Res* 1986;5(1):90–8.
- [6] Fiorini P, Shiller Z. Motion planning in dynamic environments using velocity obstacles. *Int J Rob Res* 1998;17(7):760–72.
- [7] Lacevic B, Rocco P. Kinetostatic danger field - a novel safety assessment for human-robot interaction. 2010 IEEE/RSJ international conference on intelligent robots and systems. 2010. p. 2169–74.
- [8] Lacevic B, Rocco P, Zanchettin AM. Safety assessment and control of robotic manipulators using danger field. *IEEE Trans Rob* 2013;29(5):1257–70.
- [9] Avanzini GB, Ceriani NM, Zanchettin AM, Rocco P, Bascetta L. Safety control of industrial robots based on a distributed distance sensor. *IEEE Trans Control Syst Technol* 2014;22(6):2127–40.
- [10] Flacco F, Kroger T, Luca AD, Khatib O. A depth space approach to human-robot collision avoidance. 2012 IEEE international conference on robotics and automation. 2012. p. 338–45.
- [11] Polverini MP, Zanchettin AM, Rocco P. Real-time collision avoidance in human-robot interaction based on kinetostatic safety field. 2014 IEEE/RSJ International conference on intelligent robots and systems. 2014. p. 4136–41.
- [12] Jimenez P, Thomas F, Torras C. 3D collision detection survey. *Comput Graph* 2001;25(2):269–85.
- [13] Kuffner J, Nishiwaki K, Kagami S, Kuniyoshi Y, Inaba M, Inoue H. Self-collision detection and prevention for humanoid robots. *Proceedings 2002 IEEE international conference on robotics and automation (Cat. No.02CH37292)*. 3. 2002. p. 2265–70.
- [14] Haddadin S, Suppa M, Fuchs S, Bodenmüller T, Albu-Schäffer A, Hirzinger G. Towards the robotic co-worker. In: Pradaliere C, Siegwart R, Hirzinger G, editors. *Robotics research: the 14th international symposium ISRR*. Berlin, Heidelberg: Springer Berlin Heidelberg; 2011. p. 261–82. ISBN 978-3-642-19457-3.
- [15] Haddadin S, Albu-Schäffer A, Luca AD, Hirzinger G. Collision detection and reaction: a contribution to safe physical human-robot interaction. 2008 IEEE/RSJ international conference on intelligent robots and systems. 2008. p. 3356–63.
- [16] Taubig H, Bauml B, Frese U. Real-time swept volume and distance computation for self collision detection. 2011 IEEE/RSJ international conference on intelligent robots and systems. 2011. p. 1585–92.
- [17] Taubig H, Bauml B, Frese U. Real-time continuous collision detection for mobile manipulators - a general approach. 2012 12th IEEE-RAS international conference on humanoid robots (Humanoids 2012). 2012. p. 461–8.
- [18] IEC EN 60204 2006 - Safety of machinery. Electrical equipment of machines – General requirements.
- [19] ISO 10218. Robots for industrial environments – Safety requirements – part 1: robot. 2011.
- [20] ISO 15066. Robots and robotic devices – safety requirements for industrial robots – collaborative operation. 2015.
- [21] Marvel JA. Performance metrics of speed and separation monitoring in shared workspaces. *IEEE Trans Autom Sci Eng* 2013;10(2):405–14.
- [22] Ricardez GAG, Yamaguchi A, Takamatsu J, Ogasawara T. Asymmetric velocity moderation for human-safe robot control. *Adv Rob* 2015;29(17):1111–25.
- [23] Ragaglia M, Zanchettin AM, Bascetta L, Rocco P. Accurate sensorless lead-through programming for lightweight robots in structured environments. *Robot Comput Integr Manuf* 2016;39:9–21.
- [24] Zanchettin AM, Rocco P. Path-consistent safety in mixed human-robot collaborative manufacturing environments. 2013 IEEE/RSJ international conference on intelligent robots and systems. 2013. p. 1131–6.
- [25] Zanchettin AM, Ceriani NM, Rocco P, Ding H, Matthias B. Safety in human-robot collaborative manufacturing environments: metrics and control. *IEEE Trans Autom Sci Eng* 2016;13(2):882–93.
- [26] Ragaglia M, Zanchettin AM, Rocco P. Safety-aware trajectory scaling for human-robot collaboration with prediction of human occupancy. 2015 international conference on advanced robotics (ICAR). 2015. p. 85–90.
- [27] Bascetta L, Ferretti G, Rocco P, Ardö H, Bruyninckx H, Demeester E, et al. Towards safe human-robot interaction in robotic cells: An approach based on visual tracking and intention estimation. 2011 IEEE/RSJ international conference on intelligent robots and systems. 2011. p. 2971–8.
- [28] Ragaglia M, Bascetta L, Rocco P. Multiple camera human detection and tracking inside a robotic cell an approach based on image warping, computer vision, k-d trees and particle filtering. 2014 11th international conference on informatics in control, automation and robotics (ICINCO). 02. 2014. p. 374–81.
- [29] Yamane K, Nakamura Y. Dynamics filter - concept and implementation of online motion generator for human figures. *IEEE Trans Rob Autom* 2003;19(3):421–32.
- [30] Ragaglia M, Bascetta L, Rocco P. Detecting, tracking and predicting human motion inside an industrial robotic cell using a map-based particle filtering strategy. 2015 international conference on advanced robotics (ICAR). 2015. p. 369–74.
- [31] Balan L, Bone GM. Real-time 3d collision avoidance method for safe human and robot coexistence. 2006 IEEE/RSJ international conference on intelligent robots and systems. 2006. p. 276–82.
- [32] Mainprice J, Berenson D. Human-robot collaborative manipulation planning using early prediction of human motion. 2013 IEEE/RSJ international conference on intelligent robots and systems. 2013. p. 299–306.
- [33] Ragaglia M, Bascetta L, Rocco P, Zanchettin AM. Integration of perception, control and injury knowledge for safe human-robot interaction. 2014 IEEE international conference on robotics and automation (ICRA). 2014. p. 1196–202.
- [34] Morato C, Kaipa KN, Zhao B, Gupta SK. Toward safe human robot collaboration by using multiple kinects based real-time human tracking. *J Comput Inf Sci Eng* 2014;14(1):011006.
- [35] Dietz T, Verl A. Simulation of the stopping behavior of industrial robots using a complementarity-based approach. 2011 IEEE/ASME international conference on advanced intelligent mechatronics (AIM). 2011. p. 428–33.
- [36] Gilbert EG, Johnson DW, Keerthi SS. A fast procedure for computing the distance between complex objects in three-dimensional space. *IEEE J Rob Autom* 1988;4(2):193–203.
- [37] Arechavaleta G, Laumond JP, Hicheur H, Berthoz A. An optimality principle governing human walking. *IEEE Trans Rob* 2008;24(1):5–14.
- [38] Mombaur K, Laumond JP, Yoshida E. An optimal control model unifying holonomic and nonholonomic walking. *Humanoids 2008 - 8th IEEE-RAS international conference on humanoid robots*. 2008. p. 646–53.
- [39] Zanchettin AM, Rocco P, Bascetta L, Symeonidis I, Peldschus S. Kinematic analysis and synthesis of the human arm motion during a manipulation task. 2011 IEEE international conference on robotics and automation. 2011. p. 2692–7.
- [40] Zanchettin AM, Bascetta L, Rocco P. Achieving humanlike motion: resolving redundancy for anthropomorphic industrial manipulators. *IEEE Rob Autom Mag* 2013;20(4):131–8.
- [41] Lenarcic J, Umek A. Simple model of human arm reachable workspace. *IEEE Trans Syst Man Cybern* 1994;24(8):1239–46.
- [42] Kalman RE. A new approach to linear filtering and prediction problems. *Trans ASME J Basic Eng* 1960;82(Series D):35–45.
- [43] Groote FD, Laet TD, Jonkers I, Schutter JD. Kalman smoothing improves the estimation of joint kinematics and kinetics in marker-based human gait analysis. *J Biomech* 2008;41(16):3390–8.
- [44] Sinopoli B, Schenato L, Franceschetti M, Poolla K, Jordan MI, Sastry SS. Kalman filtering with intermittent observations. *IEEE Trans Automat Contr* 2004;49(9):1453–64.
- [45] Simon D, Chia TL. Kalman filtering with state equality constraints. *IEEE Trans Aerosp Electron Syst* 2002;38(1):128–36.
- [46] Simon D, Simon DL. Kalman filtering with inequality constraints for turbofan engine health estimation. *IEE Proc Control Theory Appl* 2006;153(3):371–8.
- [47] Schutter JD, Laet TD, Rutgeerts J, Decre W, Smits R, Aertbelien E, et al. Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty. *Int J Rob Res* 2007;26(5):433–55.
- [48] Aertbelien E, Schutter JD. etal/et: A constraint-based task specification language and robot controller using expression graphs. 2014 IEEE/RSJ International Conference on Intelligent robots and systems. 2014. p. 1540–6.
- [49] Durrant-Whyte H, Roy N, Abbeel P. Real-time prioritized kinematic control under inequality constraints for redundant manipulators. MIT Press; 2012. p. 145–52. ISBN 9780262305969
- [50] Zanchettin AM, Rocco P. Motion planning for robotic manipulators using robust constrained control. *Control Eng Pract* 2017;59(Supplement C):127–36.
- [51] Prete AD, Nori F, Metta G, Natale L. Prioritized motion-force control of constrained fully-actuated robots: task space inverse dynamics. *Rob Auton Syst* 2015;63(Part 1):150–7.
- [52] Mansard N, Khatib O, Kheddar A. A unified approach to integrate unilateral constraints in the stack of tasks. *IEEE Trans Rob* 2009;25(3):670–85.
- [53] Escande A, Mansard N, Wieber PB. Fast resolution of hierarchized inverse kinematics with inequality constraints. 2010 IEEE international conference on robotics and automation. 2010. p. 3733–8.
- [54] Rocchi A, Hoffman EM, Caldwell DG, Tsagarakis NG. Opensot: a whole-body control library for the compliant humanoid robot coman. *Robotics and automation (ICRA)*, 2015 IEEE international conference on. IEEE; 2015. p. 1093–9.
- [55] Ceriani NM, Zanchettin AM, Rocco P. Collision avoidance with task constraints and kinematic limitations for dual arm robots. Cham: Springer International Publishing; 2016. p. 1285–99. ISBN 978-3-319-08338-4
- [56] Ceriani NM, Zanchettin AM, Rocco P, Stolt A, Robertsson A. Reactive task adaptation based on hierarchical constraints classification for safe industrial robots. *IEEE/ASME Trans. Mechatron.* 2015;20(6):2935–49.
- [57] Ferreau HJ, Kirches C, Potschka A, Bock HG, Diehl M. qpOASES: a parametric active-set algorithm for quadratic programming. *Math Program Comput* 2014;6(4):327–63.
- [58] Ferreau HJ, Bock HG, Diehl M. An online active set strategy to overcome the limitations of explicit mpc. *Int J Robust Nonlinear Control* 2008;18(8):816–30.
- [59] Chetverikov D, Svirok D, Stepanov D, Krsek P. The trimmed iterative closest point algorithm. Object recognition supported by user interaction for service robots. Vol. 3. 2002. p. 545–8.
- [60] Blomdell A, Dressler I, Nilsson K, Robertsson A. Flexible application development and high-performance motion control based on external sensing and reconfiguration of ABB industrial robot controllers. *ICRA 2010 workshop on innovative robot control architectures for demanding (Research) applications*. 2010.
- [61] Dagum L, Menon R. Openmp: an industry standard api for shared-memory programming. *IEEE Comput Sci Eng* 1998;5(1):46–55.
- [62] Gould N, Toint PL. Preprocessing for quadratic programming. *Math Program* 2004;100(1):95–132.